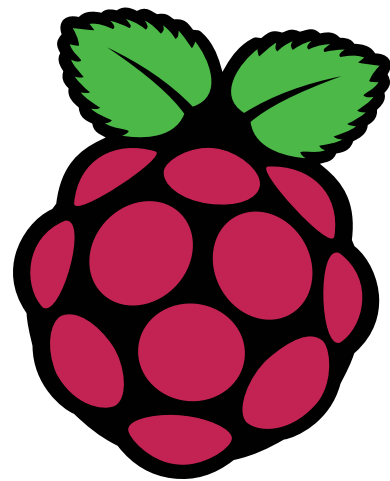




BUY IN PRINT **WORLDWIDE** [MAGPI.CC/STORE](https://magpi.cc/store)

The MagPi



Issue 109 | September 2021 | magpi.cc

The official Raspberry Pi magazine

Raspberry Pi 4
SID Synth

Upcycling a
Game Boy

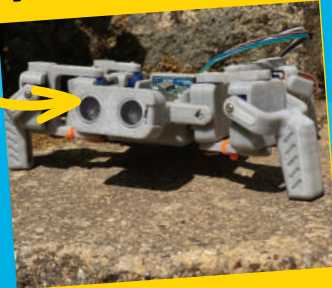
JOYSTICK



MIDI FIGHTER

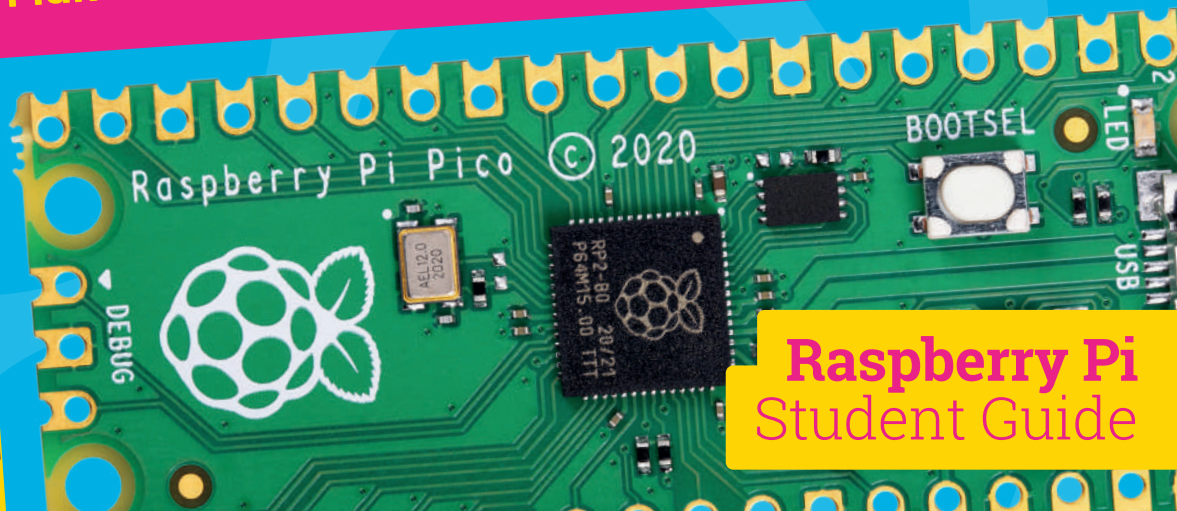


PICO ROBOT



RASPBERRY PI 20 PICO PROJECTS

Make incredible objects with microcontroller boards!



**Raspberry Pi
Student Guide**

42 PAGES OF PROJECTS & TUTORIALS



G-MASTER™
#MONITORS4GAMERS
by **iiyama**

RED EAGLE™

**QHD or Full HD?
Go CURVED!**



AMD
FreeSync
Premium

G-Master GB3266QSU
Red Eagle



G-Master G2466HSU
Red Eagle

Immerse yourself in the game with the new **Red Eagle** #monitors4gamers.
Choose the height adjustable **GB3266QSU** with a 2560x1440 resolution or the **G2466HSU** with 165Hz refresh rate and enjoy stutter-free gameplay on these 1500R curved VA panels.



Find your match at gmaster.iiyama.com

iiyama

WELCOME

to *The MagPi* 109

Building a microcontroller board took Raspberry Pi into a whole new direction. When Raspberry Pi Pico first came out, we were amazed by all the potential, but a little lost by the newness of it all. We worked with Raspberry Pi to publish a series of starter projects, and our team of talent rose to the occasion: building everything from smart guitars to games controllers.

It's been over half a year since then and we're ready to start talking about serious Pico projects. In fact, we're even discussing our favourite makes so far. See 20 Raspberry Pi Pico projects (page 30).

It's incredible what you can make with Raspberry Pi, and it's also incredible just how good Raspberry Pi is as a general-purpose computer. Our Raspberry Pi Student Guide (page 64) is packed with advice on using Raspberry Pi in a learning environment. I've been using Raspberry Pi 400 daily since it launched, and it really is astonishing how much power you get from the little board. You really don't need an expensive PC any more.

Lucy Hattersley Editor



EDITOR Lucy Hattersley

Lucy is editor of *The MagPi* and is learning Java again, for her sins. It really is a circumlocutory language isn't it?

@LucyHattersley

GET A
**RASPBERRY PI
ZERO W KIT**
WITH A SUBSCRIPTION!
PAGE 28





The **MagPi**

HackSpace
TECHNOLOGY IN YOUR HANDS

CustomPC

3 ISSUES FOR £10



FREE BOOK



magpi.cc/freebook

Subscribe to The MagPi, HackSpace magazine, or Custom PC. Your first three issues for £10, then our great value rolling subscription afterwards. Includes a free voucher for one of five fantastic books at store.rpiexpress.cc/collections/latest-bookazines
UK only. Free delivery on everything.

Contents

► Issue 109 ► September 2021

Cover Feature

30 20 Pico projects

Regulars

- 92 Your Letters
- 97 Next Month
- 98 The Final Word

Project Showcases

- 10 Automated Model Railroad
- 14 RFID Floppy Disk Reader
- 16 Remote agricultural monitoring
- 18 DMGPlus
- 20 RACoon waste sorter
- 22 HIIT Workout Trainer
- 24 Synth6581
- 26 Automated Standing Desk



30



24



18

DMGPlus

Synth6581

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

Tutorials

- 39** Ultimate home server - part 2
- 44** Galaxian attack patterns
- 47** Create GUIs in Python - part 9
- 54** Pico communication protocols
- 58** Isomorphic keyboard - part 2

The Big Feature



64

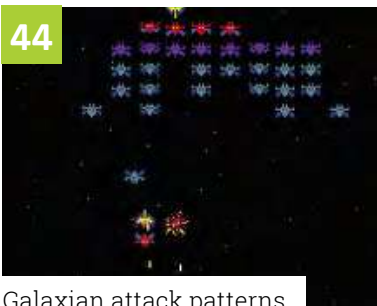
Student guide

Reviews

- 74** Pico Wireless Pack
- 76** HyperPixel 2.1 Round
- 78** Air Quality Kit
- 80** 10 Amazing game dev projects
- 82** Learn Python

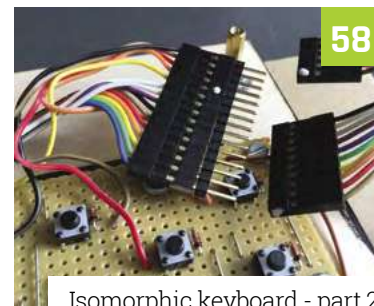
Community

- 84** Geeky Faye interview
- 86** This Month in Raspberry Pi



44

Galaxian attack patterns



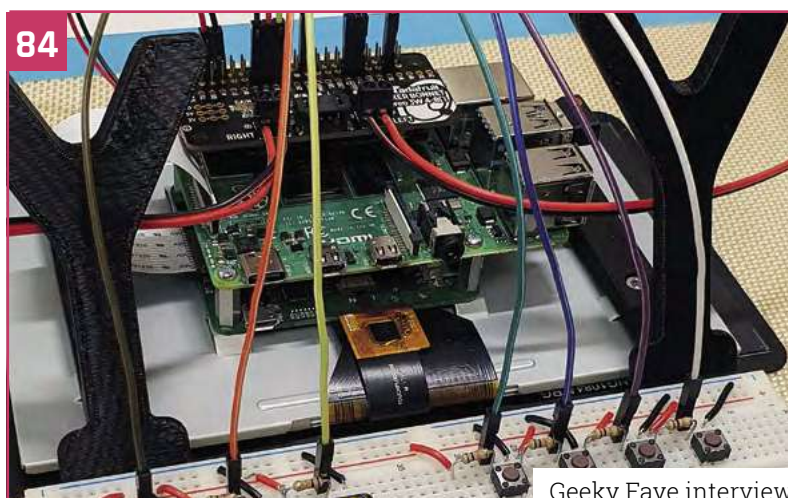
58

Isomorphic keyboard - part 2



80

10 amazing game dev projects



84

Geeky Faye interview

WIN
1 OF 10 RASPBERRY PI
CASE FANS



95

Convert Raspberry Pi 4 to a tablet with **RasPad 3.0**

- Fully compatible with Raspberry Pi 4
- 10.1" IPS touchscreen with an impressive resolution of 1280x800 pixels
- Full access to all Raspberry Pi GPIO, Ethernet, HDMI, audio, USB, and power ports
- Compatible with Raspberry Pi OS, Arduino, Ubuntu Desktop, Chromium OS, Android and more
- 10-point multi-touch screen, 5-hour battery life
- Easy to assemble



Buy now! raspad.com/products/raspadv3



Raspberry Pi 4 refurbishment programme launched

Trade-in programme enables Raspberry Pi computers to be reused. By **Lucy Hattersley**

OKdo and Sony Pi have worked together on a new scheme to renew Raspberry Pi computers. Raspberry Pi owners can send pre-loved Raspberry Pi 3 and Raspberry Pi 4 boards to OKdo Renew in exchange for £10 vouchers. Raspberry Pi boards will be sent directly to the Sony Technology Centre where they were originally manufactured.

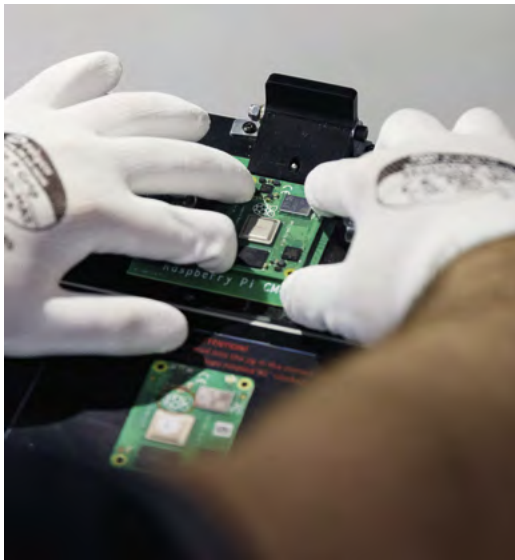
These pre-loved boards will be tested, reconditioned, and renewed. OKdo plans to put the restored boards back on sale at a lower price, and will donate to the Raspberry Pi Foundation for each board sold.

OKdo's SVP of Technology Richard Curtin comments: "With the global population predicted to approach nine billion people by 2030, we are using more resources than the planet can provide; our future depends on reusing what we have in a sustainable way."



▲ Raspberry Pi computers consume less energy, material, and labour than traditional PCs, making them inherently better for the environment

▼ Refurbished Raspberry Pi computers will be tested at the Sony Technology Centre



Raspberry Pi planet

"Raspberry Pi has a long-standing commitment to sustainability," says Eben Upton, Founder & CEO at Raspberry Pi. "Our computers consume less power than the legacy PCs they replace, and of course consume less energy, material, and labour during the manufacturing process. Over 40 million Raspberry Pi computers have been sold since 2012, and the vast majority of these remain in working order, even after their owners have upgraded to more recent versions. Managed reuse of these surplus units provides an avenue to further reduce our environmental footprint, and we welcome this exciting new initiative from OKdo and Sony."

"Our future depends on reusing what we have in a sustainable way"

"The scheme will aim to begin the recycling of a proportion of the 40 million Raspberry Pi [boards] in circulation today that are pre-loved, but no longer used," explains Richard Curtin. Initially, this service is being rolled out across the UK, but we have plans to expand it globally by the end of 2021." More information on the scheme and how to take part can be found at: okdo.com/raspberry-pi-renew.



Served for you

Electronics housings – now with displays and keypads

Electronics housings from Phoenix Contact are now available with integrated touch displays or displays with membrane keypads. You configure your customised housing solution and we take care of everything else; from printing to mechanical processing up to the pre-assembly.



For additional information call 01952 681700 or visit phoenixcontact.com/enclosures-with-displays



▲ Each time the train passes the sensor on the track, its speed is adjusted by the Pico program

Pico-Controlled Automated Model Railroad



Kushagra Keshari

Kushagra Keshari has always been fascinated by electronics and is halfway through a degree in mechatronic engineering in India.

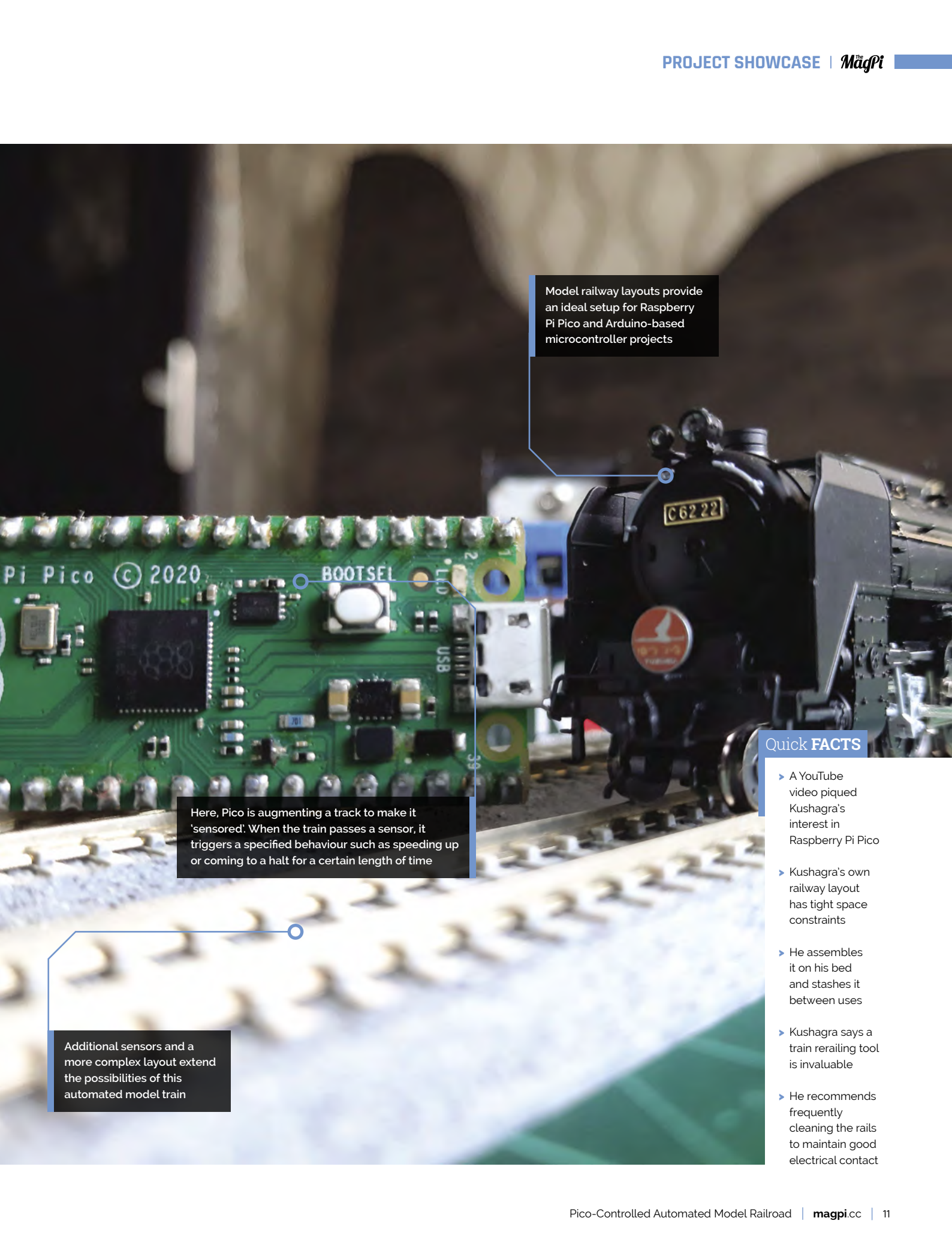
magpi.cc/kushagra

Automation and model railways prove a great coupling for one enthusiast. **Rosie Hattersley** tracks its progress

Who doesn't love a model railway? Mechatronics student Kushagra Keshari is a big fan, and often combines his love of locomotion with his interest in electronics. Having spotted a YouTube video introducing Raspberry Pi Pico, he decided it was high time he investigated the microcontroller and explored how it might be incorporated into his model railway setup. The result, Pico-Controlled Automated Model Railroad (magpi.cc/picorailroad), builds on his 'sensored' track concept that detects a train passing and instructs it to speed up, slow down, or come to a halt. "Electric motors are among my most favourite devices because they connect the fields of electronics and mechanics," he reveals. "I try to add motors wherever possible."

As someone with experience of Arduino, Kushagra believes Pico has great crossover appeal, lauding its ease of use and functionality: "One of the best reasons to [use Pico] is the availability of libraries to allow the integration of modules, sensors, and other complex stuff without having to write the entire code from scratch," he explains in his guide to programming Raspberry Pi Pico with the Arduino IDE (magpi.cc/picowitharduinoide).

The project joins Kushagra's sizable collection of Instructables focused on using microcontrollers and stepper motors to augment a model railway layout. His projects work well together, with this first foray into Pico territory building on Kushagra's existing ideas for using Arduino to



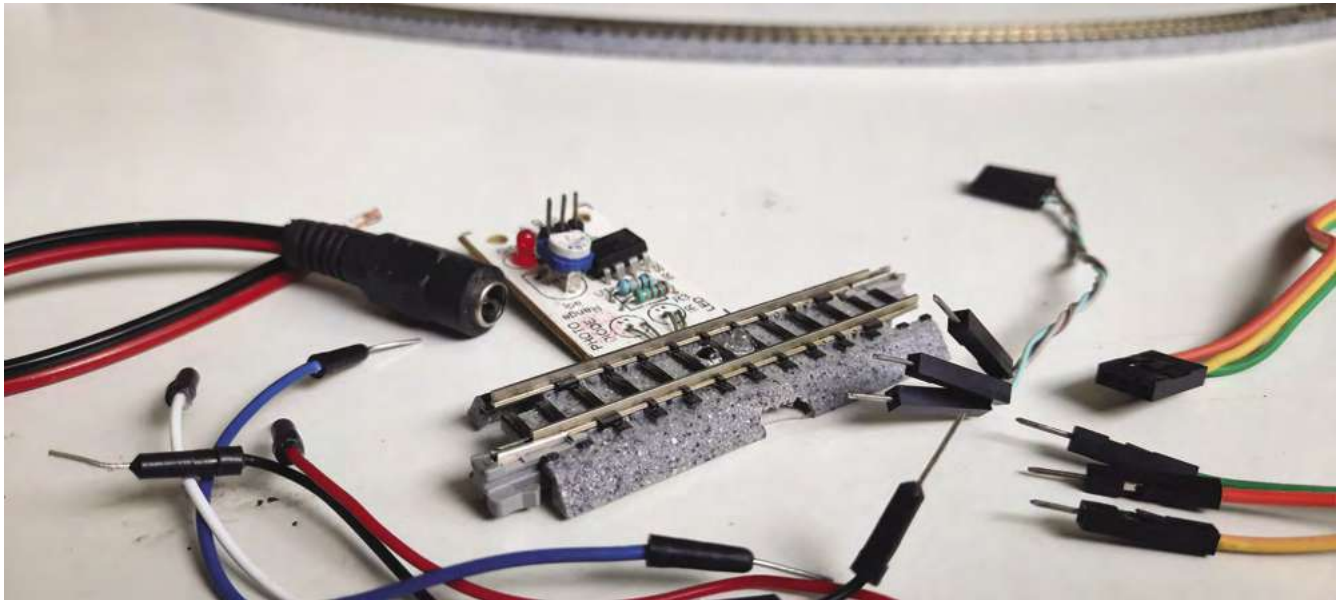
Model railway layouts provide an ideal setup for Raspberry Pi Pico and Arduino-based microcontroller projects

Here, Pico is augmenting a track to make it 'sensored'. When the train passes a sensor, it triggers a specified behaviour such as speeding up or coming to a halt for a certain length of time

Additional sensors and a more complex layout extend the possibilities of this automated model train

Quick FACTS

- ▶ A YouTube video piqued Kushagra's interest in Raspberry Pi Pico
- ▶ Kushagra's own railway layout has tight space constraints
- ▶ He assembles it on his bed and stashes it between uses
- ▶ Kushagra says a train rerailing tool is invaluable
- ▶ He recommends frequently cleaning the rails to maintain good electrical contact



▲ An IR proximity sensor is used to detect the train passing on the track

add smart features to his train tracks. “Raspberry Pi Pico helped make it more budget-friendly and enabled several options for expansion in the future,” he says.

Making tracks

Kushagra’s approach to automating a model railway starts with a basic oval track layout, Raspberry Pi Pico, and an L298N motor driver, which he favours because of its $\pm 5V$ on-board voltage regulator, which can be used to supply power to any attached microcontroller and other modules. “The voltage level is quite important as a lot of sensors and modules use 3.3 volts for communication, and having a 3.3-volt microcontroller eliminates the need of using voltage level shifters and optocouplers in the circuit,” he explains.

▼ Raspberry Pi Pico reads the track sensor and controls the L298N motor driver connected to the track

Raspberry Pi Pico is programmed using the Arduino IDE and becomes the brains of the model railway, using sensors to instruct each locomotive that passes over the smart track what to do next. As the layout grows, additional sensors can be added and used to alter the locomotive’s speed or get it to stop for a preset time, as if it were at a red signal or at a station. Kushagra says, “I am planning to add yard shunting operation which will allow the locomotive to couple, and decouple, with the rolling stock to allow it to pick them up from one point and drop them off at another.”

Train brain

The setup instructions given on his Instructable are admirably clear, but Kushagra also provides troubleshooting tips for this and his other projects, such as what to do if the train unexpectedly starts



▼ Layout additions will see Pico controlling the automatic running of multiple locomotives



“ Raspberry Pi Pico is programmed using the Arduino IDE and becomes the brains of the model railway ”

moving in the wrong direction (switch the polarity of the track’s power). For example, the sensor’s LED should only light up when a train passes over the sensor-enabled track. If, however, the LED on the sensor is always on, you should adjust the sensitivity of the potentiometer.

Although there are lots of other Raspberry Pi- and Arduino-controlled layouts out there, we really like Kushagra’s low-cost approach to automating his model railway. His modular approach makes it clear enough for newbie makers to take on, while also being ideal for established railway modellers to use as a template to retrofit their layout. “Given enough time and effort, you could automate switching operations, automate the operation of multiple trains at the same time, control the lighting, or use WiFi or Bluetooth to control your trains and turnouts and a lot more,” Kushagra enthuses.

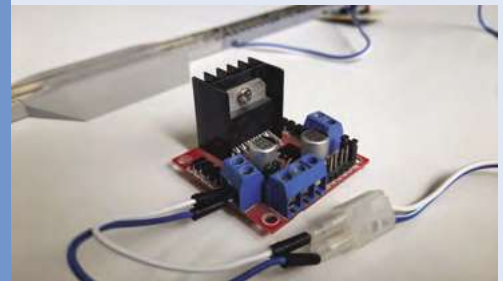
There’s plenty of potential in the ‘sensored’ track approach, too: anyone up for combining the track sensor with a station announcement board to indicate an approaching service (or delay)?

Train time

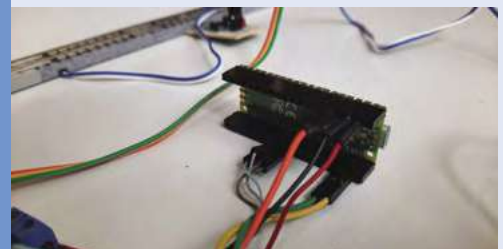
To automate your own model railway, you’ll need a Raspberry Pi Pico and an L298N motor driver allowing two DC motors to be controlled at once. You’ll also need an IR proximity sensor to make your track ‘sensored’ (see magpi.cc/sensoredtrack).



- 01** Program your Pico using the Arduino IDE code provided in Kushagra’s GitHub repo (magpi.cc/picorailroadgit), then create your track layout, ensuring the ‘sensored’ section of track is located as indicated with the higher arrow. The locomotive is positioned on the lower arrow.



- 02** Connect the wires of the track power block to the outputs of the motor driver, then connect the female DC jack positive wire to the +12V terminal. Connect a male jumper and the DC jack’s negative wire to the GND (centre) terminal, and a jumper wire to the +5V (right) terminal of the motor driver.



- 03** Attach Pico to the motor driver via the 5V pin. Connect GP7 to ENA, GP8 to IN1, and GP9 to IN2. Connect the sensed track to the Pico and the driver to a 12V adapter, then power on.

RFID Floppy Disk Reader

Ever wanted to make contactless floppy disks? Maybe not, but **Rob Zwetsloot** takes an appropriately distanced look at some



Dylan Blake

A business analyst from Texas who codes in Python and SQL for a large grocery chain.

Old-looking tech with futuristic properties is a popular concept in media these days, so much so that we're surprised this is one of the first projects like this we've seen.

"My project was taking an old 1988 word processor and repurposing it into a gaming emulator with a Raspberry Pi," creator Dylan Blake tells us. "I wanted to utilise the floppy disk drive with RFID tags to initiate the software and have a working power button for added effect."

We've covered an RFID-based record player that did something similar with vinyls, but it wasn't built into an old piece of tech like this. "I came up with the idea by realising I didn't have a cool case to put my emulator in, and I really dig all things retro," Dylan explains. "I found this device for \$20 on a marketplace app and thought it would be awesome to work with."

Breathing new life

The way the system works makes it feel almost like a classic computer, albeit a bit faster.

"When you click the tactile power button, you are briefly greeted with a retro splash screen and then cute computer ASCII art prompting to insert a floppy disk," Dylan says. "You fumble around for your favourite game handwritten on a 3.5-inch floppy, insert it into the floppy disk bay, and your game immediately starts up. If you don't know what game you want to play, you can insert the 'All' floppy to access the RetroPie game menu (of course, using the 8-bit theme)."

Dylan chose Raspberry Pi to power this for all the familiar reasons – a good size, easy access



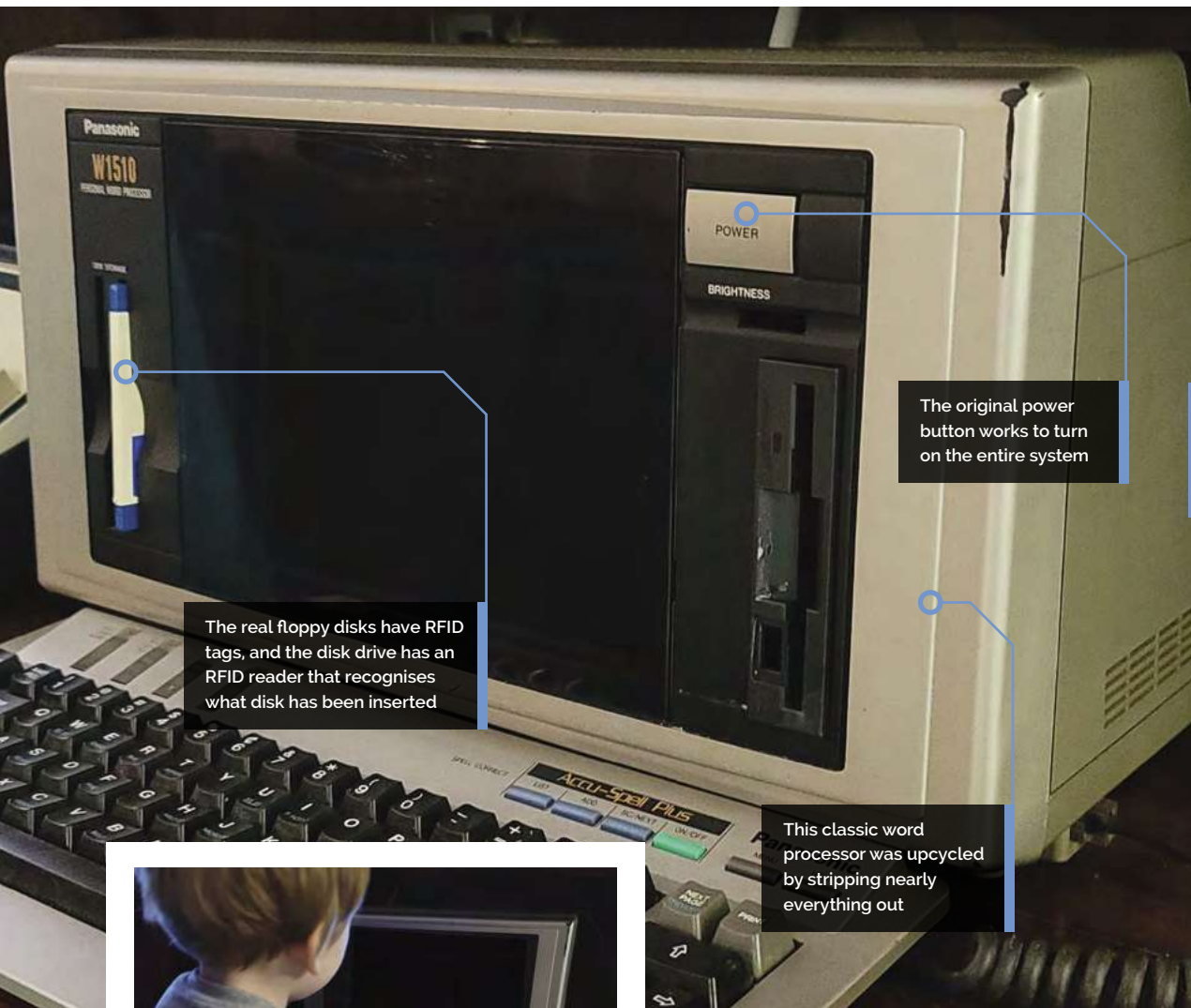
" I especially like that my two-year-old son enjoys playing with it, albeit poorly **"**

to GPIO pins, and it also allowed him to get more comfortable with Linux. "Raspberry Pi has fascinated me for years, and I probably have five of them at this point for various projects."

Old-school cool

We're big proponents of learning things when building projects, and as well as getting more experience with Linux, Dylan learnt how to use RFID tags and readers in the process, which we think is a cool skill. It wasn't his first choice, though.

"I have a USB floppy reader that I would like to utilise instead of the RFID tag reader for look and feel," he admits. "But right now I like the ease of use of the RFID reader."



Quick FACTS

- ▶ As the printer has been removed, there's a lot of empty space on top
- ▶ Unfortunately, the original keyboard could not be used just yet
- ▶ This is because the keyboard connects via RJ45
- ▶ Word processors used to be machines, not just the name of PC software
- ▶ It uses an 800x600 display instead of the original



He says the reactions he's received from it completely validate why he did it: "It was very popular on Reddit where I originally shared it, and my family loves how niche and fun it is! I especially like that my two-year-old son enjoys playing with it, albeit poorly. The only negative feedback I've received is that I repurposed a device that was already working. My counter to that is the word processor was only good for typing documents and saving them to floppies. Repurposing it has given life to this old tech, even if it's just the aesthetics of the original device." [M](#)





Remote agricultural monitoring

Invasive species affecting crops are an issue around the world. One government agency in Japan is trying to monitor fruit flies. **Rob Zwetsloot** investigates



MAKER

Shinobu Sukenari

A Senior inspector at the Naha Plant Protection Station of the Ministry of Agriculture, Forestry and Fisheries.

mechatrax.com

Tephritid fruit flies can be a huge pest when it comes to agriculture, destroying various crops wherever they go. To try and stay ahead of any swarms, the Japanese Ministry of Agriculture, Forestry and Fisheries (MAFF) is using Raspberry Pi-powered monitoring stations.

The system was designed and programmed by Shinobu Sukenari, a MAFF senior inspector, using the following requirements: they should take a photo of the trap at least once a day and report them over the internet. Secondly, the systems should run for a month without external power. Finally, the system should be protected from the sun, wind, and rain.

Fly trapped

“We use a [trap] to capture and kill the target,” Soichi Nagasato, CEO of Mechatrax who helped develop the monitoring station, tells us. “Then we photograph the inside of the trap. The camera uses a Raspberry Pi Camera Module V2 and is connected to a Raspberry Pi 3. Other components include a power management board for intermittent

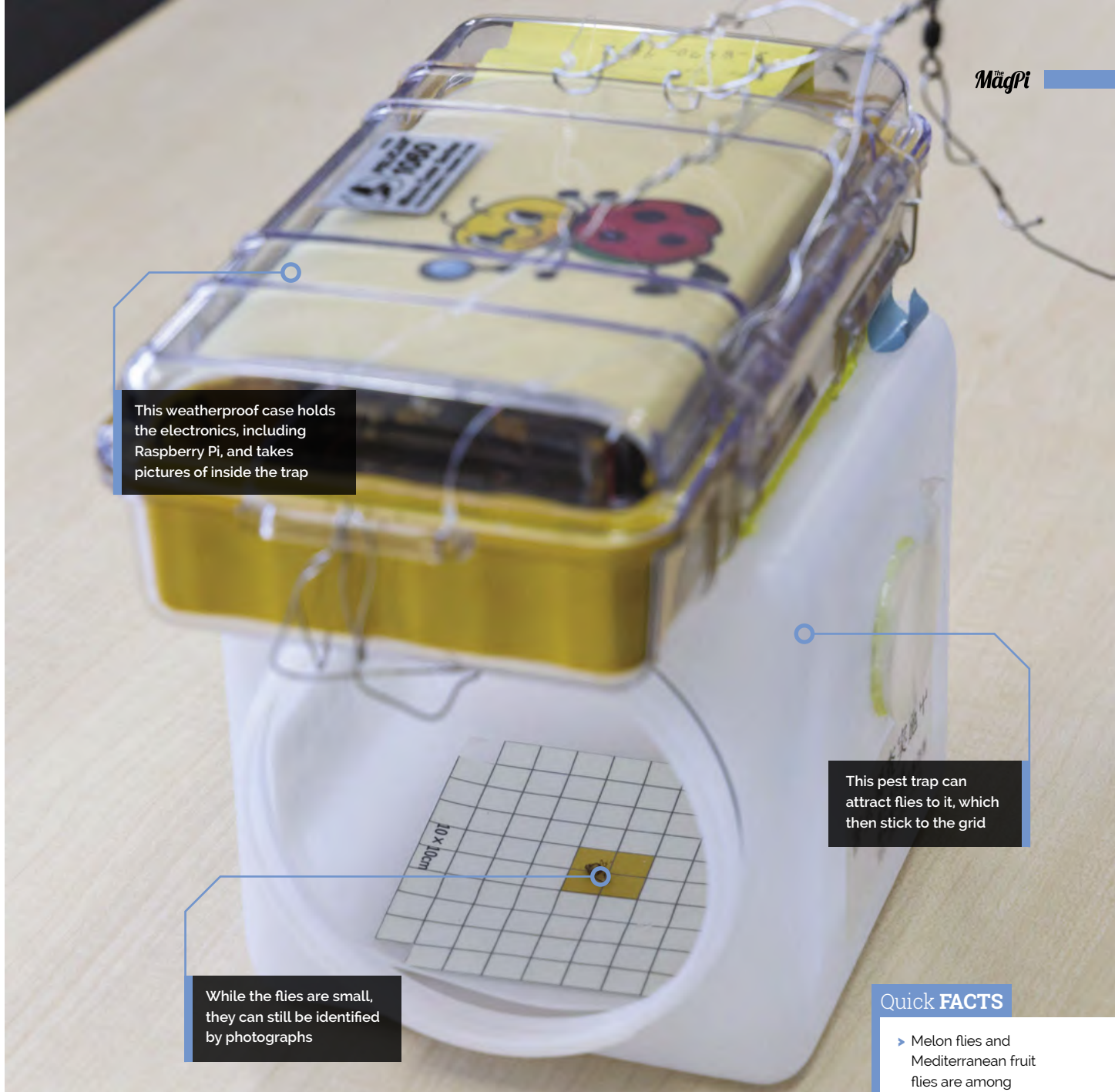
operation, an LTE [4G wireless network] board, and dry batteries. The complete set of equipment is housed in a waterproof box and attached to the pest trap. The inside of the trap is photographed through a window in the waterproof box and the images are emailed to the administrator via [the internet].”

The batteries are nothing special – just eight AAs, enough to power a Sega Game Gear – and even the special case is made from an off-the-shelf, waterproof carrying case: “We use the Pelican 1060 Micro Case,” Soichi explains. “It is a transparent case that allows photography from the inside and has no openings. As a result, Pelican’s high water and weather resistance can be used as is.”

Analysis on the fly

As you may be aware, flies are pretty small. This could make it tricky for the system; however, the results were adequate.

“Although fruit flies are roughly 1 cm in size, we were able to obtain enough clear images by email to distinguish between native and target species,” Shinobu says. “This may reduce the time and



This weatherproof case holds the electronics, including Raspberry Pi, and takes pictures of inside the trap

This pest trap can attract flies to it, which then stick to the grid

While the flies are small, they can still be identified by photographs

Quick FACTS

- Melon flies and Mediterranean fruit flies are among the target species
- Fruit flies lay their eggs in fruit; the larvae then devour the interior
- Some fruit flies are considered good for agriculture
- Okinawa has a tropical climate and is made up of over 150 islands
- Sugar cane, vegetables, and fruit are grown in Okinawa

“ Although fruit flies are roughly 1cm in size, we were able to obtain enough clear images ”

effort required for field patrols and improve the frequency of checks in areas/seasons with a high risk of infestation.”

How to drive off the unwanted flies is not part of the system. It's still a prototype, albeit one that is undergoing positive trials. However, as they are able to identify when and where they're present more quickly and efficiently than with patrols, these methods can be more effective. [M](#)



▲ Sample flies to use for checking the photos

DMGPlus

This Game Boy looks like an original machine, but Jeroen Domburg has played around with its innards and it's unlike anything we've seen before, as **David Crookes** explains



Jeroen Domburg

Aka Sprite_tm. Jeroen is a long-time maker and hacker of anything that has electrons running around in it.

spritesmods.com

▼ You can't tell the difference between the finished DMGPlus project and the original Game Boy – all the alterations are inside

The Nintendo Game Boy – the iconic handheld video game console launched in 1989 – is no stranger to the pages of *The MagPi*. We've seen makers either stuff a Raspberry Pi computer into an original case or buy off-the-shelf projects, such as the superb RetroFlag GPi, and create their own from scratch. It's great to see the device kept alive.

But just as we thought we'd seen it all, along came Jeroen Domburg, aka Sprite_tm. Like us, he'd seen a reasonable number of people modifying Game Boy cases to create portable RetroPie machines. "But because they wanted the thing to emulate as many consoles as possible, they usually went all-out with the modifications: high-resolution screen, Li-ion battery, HDMI and USB, multiple front buttons, shoulder buttons, the works," he says.

"Obviously this would work really well, but it went against the original Game Boy looks. The projects could look like a weird mutation and it

made me think, what if I went the other way? What if instead of sacrificing the original looks for playability, I sacrificed playability for the original looks?" Welcome then, DMGPlus: a handheld that looks familiar but has its internals replaced by something more powerful.

Pressing the right buttons

That something includes a Raspberry Pi Zero computer and a replacement motherboard containing a lower power, high performance ICE40 field-programmable gate array (FPGA). These are fixed either side of a new, printed circuit board, replacing the CPU, GPU, and memory.

Jeroen has retained the buttons, cartridge port, speaker, and link port, with everything capable of being run from four AA batteries, just like the original. "I did change the LCD a little bit by driving it in a smart way so that it can display 16 greys instead of the original four," he enthuses.

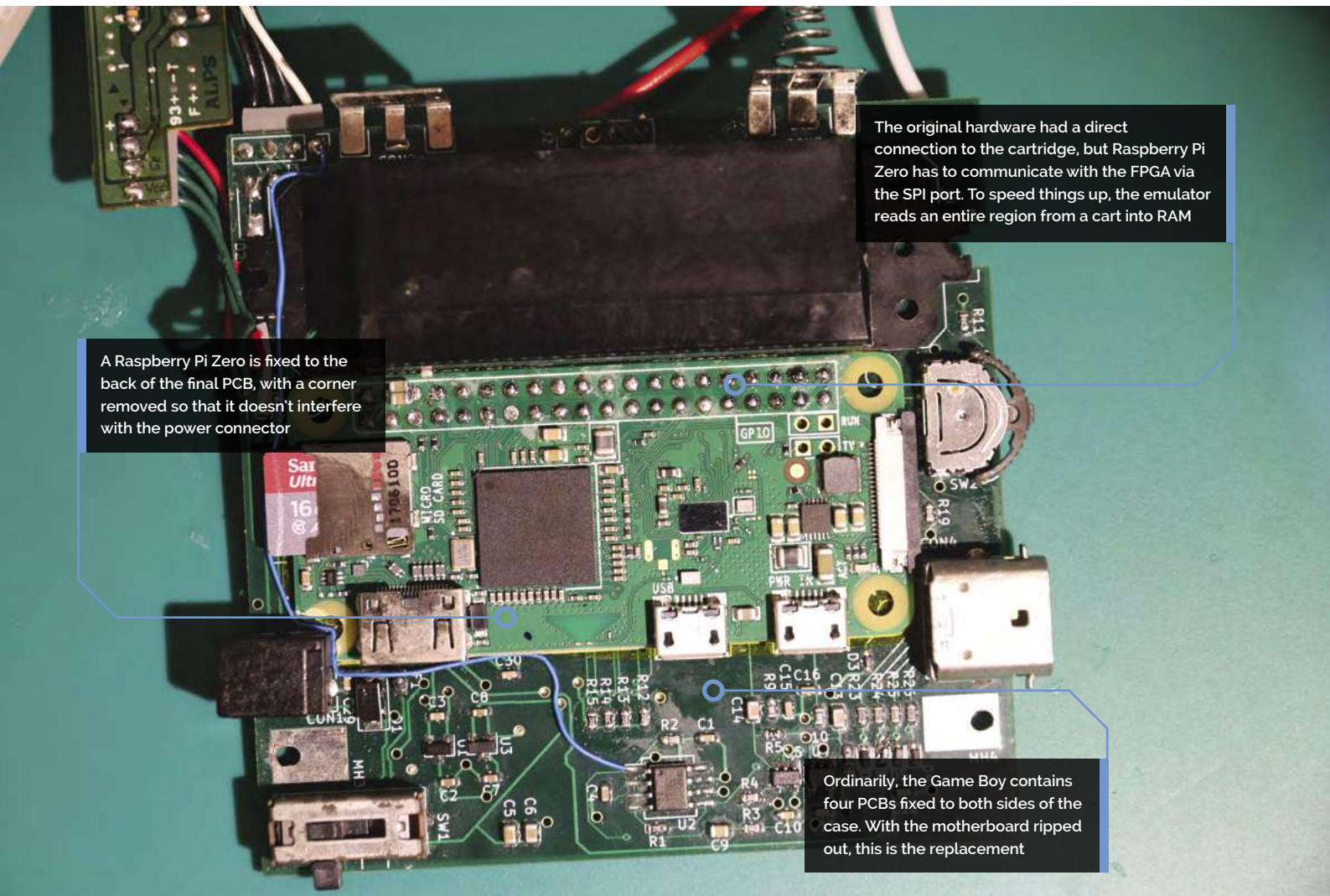
And the upshot of that? "It ends up substantially increasing the number of games the Game Boy can play," he continues. "Because of emulation, all of a sudden you can have access to games that originally ran on other consoles, some of which have specs way better than the original Game Boy."

Work hard, play hard

Making the build extra-special is its use of original carts, emulating the Game Boy experience so closely it's difficult to tell if anything has changed. It uses the emulator GnuBoy and when Jeroen uses his own reproduction carts containing games not originally made for the Game Boy, Raspberry Pi Zero kicks in and runs the title natively.

"Getting Raspberry Pi Zero to boot as fast as possible was tricky because it needed some rethinking of the boot process, as well as a kernel recompile to make it load within the time it took the Game Boy startup screen to finish," Jeroen explains. "My hardware also takes a longer path:





The original hardware had a direct connection to the cartridge, but Raspberry Pi Zero has to communicate with the FPGA via the SPI port. To speed things up, the emulator reads an entire region from a cart into RAM

A Raspberry Pi Zero is fixed to the back of the final PCB, with a corner removed so that it doesn't interfere with the power connector

Ordinarily, the Game Boy contains four PCBs fixed to both sides of the case. With the motherboard ripped out, this is the replacement

Raspberry Pi has to talk through the SPI port to the FPGA, which then needs to control the cartridge. Doing this for every byte that the game needs would be very slow, so the emulator uses caching.”

Raspberry Pi Zero seemed the perfect choice. Aside from being able to fit in the case, Jeroen said he knew he could get the video interface to do what he wanted. “Raspberry Pi has proper DPI support, outputting video over the GPIO pins so I could make the Game Boy LCD show up as just another frame buffer device,” he says. “That was important because I didn’t want to hack the video output system of every emulator or game I wanted to run it.”

The result is a stunning handheld console, but not one for the faint-hearted. “The big challenge was the need for custom hardware, custom software, custom gateway, and so on and it took a fair bit of time and effort to develop,” he says. “If you’re looking to replicate it, be prepared to put some work into tweaking and fixing things.”

“ Instead of sacrificing the original looks for playability, I sacrificed playability for the original looks ”



▲ This is a reproduction cart. Figuring how to program them properly was tricky because they’re used outside of their specified voltage range, even in a standard Game Boy

Quick FACTS

- ▶ The project took three years to develop
- ▶ The original Game Boy motherboard is replaced
- ▶ It will run games from the original carts
- ▶ The emulator runs on Raspberry Pi
- ▶ You’re not restricted to playing Game Boy titles

RACCOON

waste sorter

Students at Rice University have created a smart bin which prevents recycled waste from becoming contaminated with bad rubbish, as **David Crookes** discovers



Rene Carballo

The RACCOON waste sorter has been created by Rene Carballo, Patrick Chickey, Armando Amigon, Kevin Mejia, Angel Soto, and Kevin Tu.

magpi.cc/racoon

Recycling rates for household waste vary across the world. Germany is the most impressive, recycling 66 percent of its trash – a figure well above the EU average of 48 percent. The UK's rate is 46.2 per cent, the US is 35 percent, but some countries such as Chile and Turkey barely recycle at all.

One of the problems faced by individuals is uncertainty over what can and cannot be recycled. While recycling bins are commonplace in most countries, they can become contaminated with the wrong material and that's why, in the UK alone, some 525,000 tonnes of household waste ends up being rejected at the point of sorting.

This can prove expensive to rectify, but a team of students at Rice University in Houston, Texas, have been working on a low-cost solution. They've created a prototype waste bin that keeps a close eye on what is thrown away. "We wanted to provide a safety net for users so they can recycle without fear

of contaminating recycling loads," says one of the students, Rene Carballo.

Trash training

In designing their project, they opted to use a Raspberry Pi computer as the main control system. "We chose it because of its small but powerful form-factor," Rene says. They supplemented it with an Arduino to control some sensors and motors. "The use of these two platforms allowed us to make a device that would accurately classify and sort recyclables, while being easy to use and maintain," he explains.

The system uses a camera to take an image of any trash placed within an intake section that sits between two compartments. This is sent to Raspberry Pi which then seeks to classify the object that has been thrown away, allowing a decision to be made as to whether the item is non-recyclable or not.

If so, it is sorted into the correct bin, ensuring the one for recycling isn't contaminated. "The main decision-making mechanism is a neural network that takes an image of the recycled item in order to classify it," Rene reveals. Gathering the data took three months.

"To train the network, we had to source our own data, taking more than 1500 images by hand," he continues. "As this is a relatively small dataset for deep learning, we mitigated the issue by using transfer learning and data augmentation."

Cleaning up

As well as sorting rubbish, the device doubles as an educational tool, the idea being that users end up learning what should and shouldn't be recycled without needing assistance from the smart bin.

"The device is coupled with a companion web app which provides resources on the best recycling practices so people can better understand how to recycle," says Rene. "It also allows users to check



▲ This vertical shot of the prototype waste bin shows the box into which the rubbish is dropped and the mechanism used to move it left and right



Waste is placed through this slot and sits within a box, allowing a camera connected to Raspberry Pi to take an image

A stepper motor rolls the walls of the waste box left or right, depending on whether the trash has been classified recyclable or not

The students have created a machine learning classifier for Raspberry Pi. The images are fed to the computer

Quick FACTS

- ▶ Rice University gets paid for its recycled waste
- ▶ But 13% of it goes to landfill
- ▶ This costs the university more than \$1000/month
- ▶ This project took seven months to make
- ▶ It's been nominated for the 2021 James Dyson Award

▼ A limit switch inside the box recognises when an item has been dropped inside, and locks the door with a servo-controlled deadbolt while it's being processed

“ We had to source our own data, taking more than 1500 images by hand ”

in to the device with a personalised QR code, giving them rewards points when they recycle correctly.”

The students are now looking to refine their project, with the immediate aim of making their own university more sustainable by placing these bins around the campus. “We want to improve on the algorithm to increase the accuracy of the device, and spend more time on the web app to allow for more personalised feedback to the user,” Rene says. “It’s more important than ever to take care of our planet.”



HIIT Workout Trainer

If watching the Olympics has motivated you to improve your fitness, here's a fabulous workout idea using a Raspberry Pi. **Nicola King** feels the burn



James Wong

James is a machine learning engineer working on building industrial recommender systems and he enjoys building things with data science.

magpi.cc/hiitpi

When James Wong felt that his workout routine needed a boost due to extended periods of having to work from home, he knew he needed to take action.

“First of all, working from home all day long without going to the gym has taken a toll on my health,” he explains. “Secondly, as a machine learning practitioner, doing research on edge devices like Raspberry Pi and microcontrollers has always been fascinating to me, in the sense that the computational efficiency continuously pushes the boundaries of what we can achieve on a \$30 computer with ML.”

With that in mind, James married his regular HIIT (high-intensity interval training) workout with Raspberry Pi, and the HIIT Raspberry Pi web app was born. It's a clever idea that uses

machine learning on Raspberry Pi to keep tabs on your workout in real time and make sure you are getting optimum results.

No pain, no gain

James first had the idea in January 2020, and had a prototype ready around three weeks later, with improvements and tweaks then made based on feedback from family, friends, and internet reaction. He's taken the HIIT concept, where you have short bursts of very high-intensity activity alternated with rest or much lower intensity exercise, and arguably improved it.

“It gamifies workouts with the leaderboard dashboard, and makes doing exercise fun with your partner and friends,” he says. What's more, the app is simplicity itself in that it basically does just two things. Firstly, it uses computer vision, and a Raspberry Pi Camera Module, to track movements and poses, and then it scores them based on a set of predefined standards.

To speed up the machine learning algorithms, James plugged a USB Coral Accelerator Edge TPU into his Raspberry Pi. This, he highlights, was an important addition: “An Edge TPU is required for it to work roughly at 30 fps. Without the accelerator, frame rates drop significantly and the user experience deteriorates.”

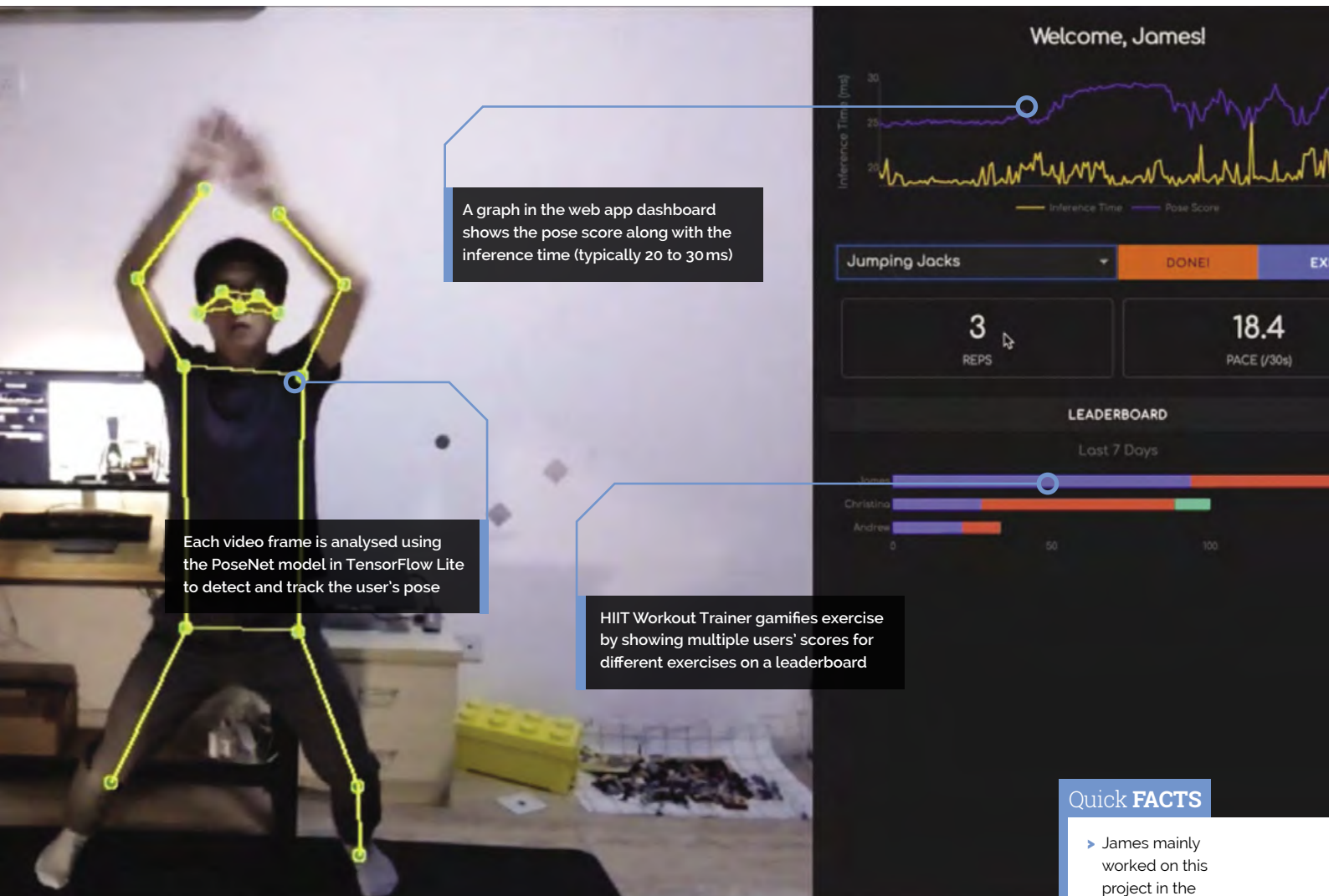
Of course, as with all builds, some fine-tuning was required, as James shares. “Every single workout consists of many moves or a sequence of poses. HIIT Workout Trainer needs to make sure to track them at a fraction of a second under various external conditions. Also, the web app interface has gone through many evolutions to have an intuitive and simplistic version of high-performance.”

James has made various improvements to the project, with most coming from faster and more efficient video processing by the ML model under the hood, as well as user interface redesigns on the front. “I'm definitely going to test more workouts

“ It gamifies workouts with the leaderboard dashboard, and makes doing exercise fun with your partner and friends ”



▲ The Coral USB Accelerator's Edge TPU processor enables Raspberry Pi to analyse images at 30 frames per second



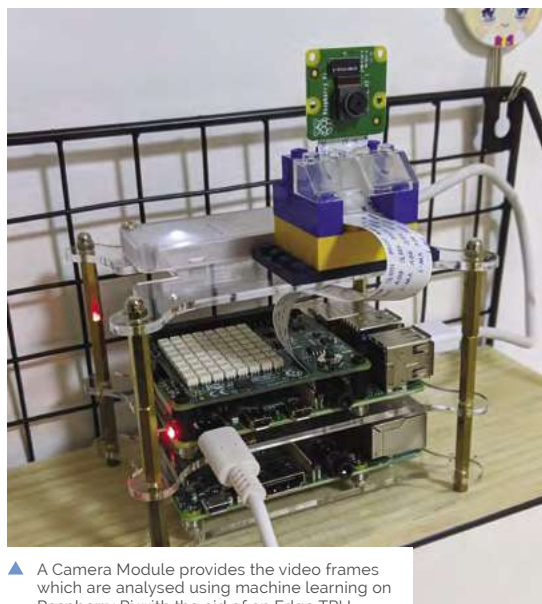
if I have enough time on hand. It would be great if more people joined in,” he says.

Unbiased adjudicator

This idea has obvious potential. James has described the app as akin to an “electronic referee”, and it could easily be adapted for other forms of sport or exercise: “I’ve already seen people do similar awesome projects, but for weight training.”

So, if you’d like to try your hand at this project, James very much encourages it. “The project is open-sourced on GitHub [magpi.cc/hiitpigit], where you can find more to set it up and get started. If anyone has any questions, please find me via email or Twitter [see magpi.cc/jameswong], I’m more than happy to help.”

What more motivation do you need? As James reminds us on his web page, “Just imagining a workout is never the same as actually doing it. With everything put into place, let’s slip on sweatpants and get it rolling!” [📄](#)





Synth6581

▲ Knobs and sliders on the keyboard are used to alter SID sound parameters such as frequencies, waveforms, filters, and modulation

Using a stack of SID chips, literally, Simon Martin has recreated the classic synth sounds of the Commodore 64. **Phil King** gets all nostalgic



Simon Martin

As senior principal engineer at Raspberry Pi, Simon is the designer of Raspberry Pi 400 and the High Quality Camera. In his spare time, he likes to build and collect synthesizers.

magpi.cc/simonmartin

While he admits it doesn't look much like a musical instrument, Simon Martin's Synth6581 is a wonderful tribute to the sounds of the Commodore 64, the best-selling 8-bit computer of the 1980s.

"I always loved the sound of the Commodore 64 when I was a teenager," he tells us. "I used to 'poke' registers in the MOS6581 SID chip inside the C64 to create strange sound effects."

The C64 was limited to a single SID sound chip and Simon could only program in BASIC at the time, but he thought that one day it would be great to make a MIDI-controlled synthesizer out of it. "Delving deeper, I found that the designer of the SID chip, Bob Yannes, also had ideas of making a polyphonic synthesizer out of it but it never happened. I thought I would make it happen after all."

Authentic audio

Simon realised his polyphonic synth would have a rich, fat sound if he dedicated one SID chip per channel, and so six chips would make it possible to play six notes at once. "I wanted to make it sound like an original C64 computer and so using eighties silicon was going to be the way [rather than emulating sounds]."

Sourcing six original MOS6581 SID chips on eBay, he designed a PCB to accommodate each chip. After hand-assembling the six board units, he stacked them on top of a Raspberry Pi 4 which 'pokes' their registers to control them.

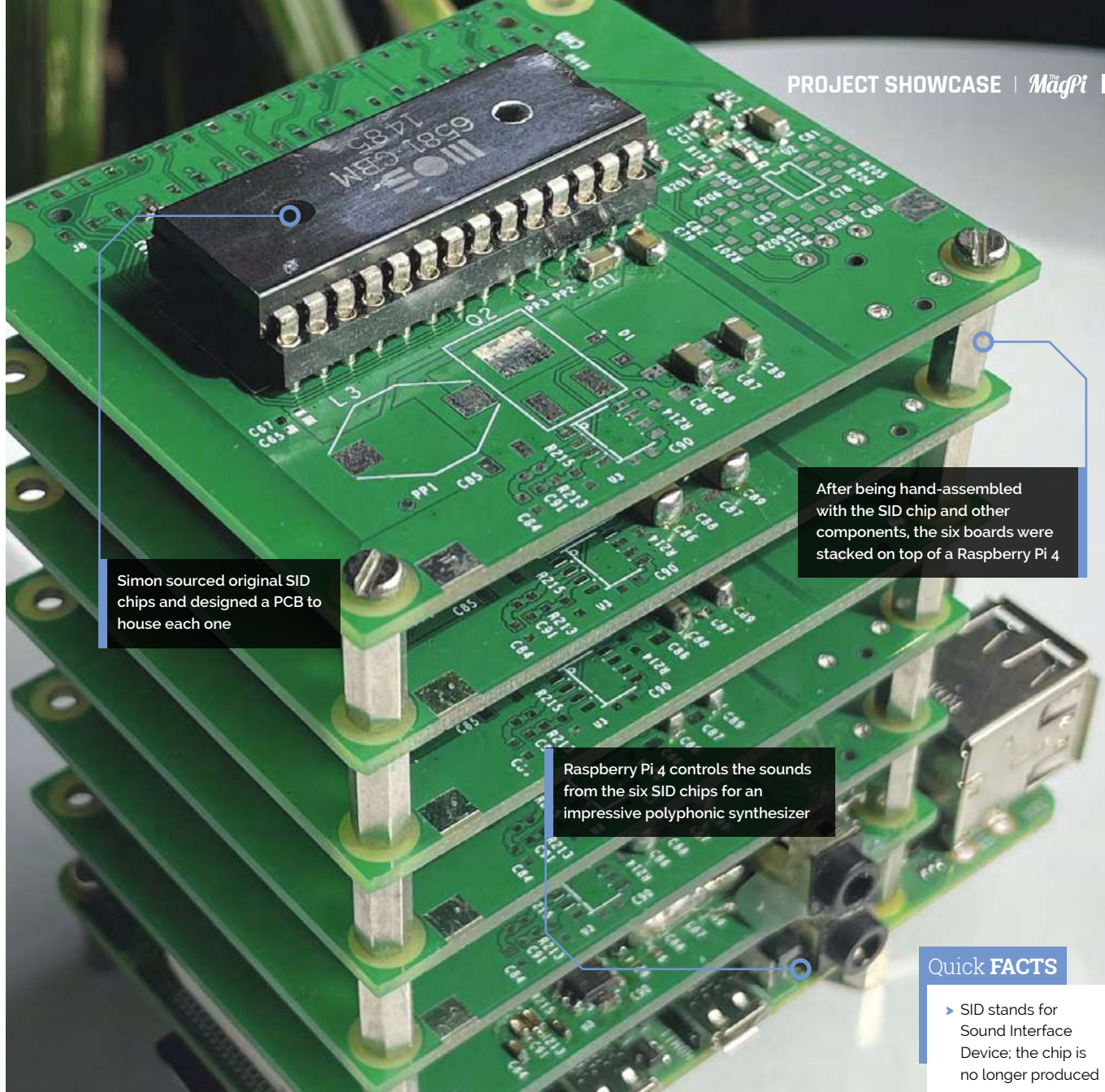
From a PC-based software synthesizer he'd written in Python some years ago, Simon ported the MIDI control part of the code to Raspberry Pi and modified it to control external circuit boards.

The biggest challenge was getting the SID chips to make any sound at all. "I wrote a C-driver that manages the I/O to control the SID chips because Python was slow at dealing with I/O," he says. "Since the filter of each SID chip has different characteristics, I had to plot the cut-off frequency vs filter setting of each channel and make a look-up table so that the chips behaved the same way."

One downside of using multiple SID chips is how much background hiss they make. "When you have six chips mixed to one output and you are not playing anything, the chips all make low volume whirring noises out the audio output. The board design can go up to eight SID chips, but that is even more background noise."

Depeche modes

The Synth6581 offers two main modes. In the standard synthesizer mode, audio from the SIDs is played using a standard MIDI keyboard connected to Raspberry Pi via USB. As shown in Simon's YouTube demo ([magpi.cc/synth6581demo](https://www.youtube.com/watch?v=9m8j8j8j8j)), the keyboard's sliders and knobs are used to tweak various parameters of the SIDs' voices. Filters can also be added so sounds evolve over time, while arpeggiator settings include the ability to modulate one oscillator to recreate that familiar ultra-fast arpeggio sound used in so many C64 tunes.



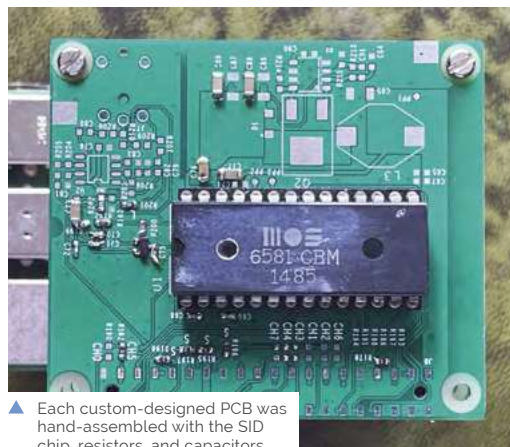
Quick FACTS

- ▶ SID stands for Sound Interface Device; the chip is no longer produced
- ▶ Simon's favourite C64 songs are Cybernoid and Ocean Loader 3 – Firefly
- ▶ He has created Synth6581 demos of classic tunes: magpi.cc/simonmartinyt
- ▶ The project took 250 hours of work, 60 hours of it on the hardware
- ▶ The code and schematics are available on GitHub: magpi.cc/synth6581gh

“I wanted to make it sound like an original C64 computer and so using eighties silicon was going to be the way”

The second main mode for Synth6581 is music playback. This enables the playing of distinctive sounds reverse-engineered from music files of C64 games. A single key press can trigger playback of elements such as percussion loops and basslines.

The public response to Synth6581 has been very positive. “People have thanked me for reminding them of games played 30 years ago and enjoyed the reboot of those sounds,” says Simon. While he plans to make further improvements to it, he also has an eye on other musical projects. “One really big challenge would be to open up an old Roland synthesizer and use the Raspberry Pi to control its sound registers directly. It’s a great hobby, but very time-consuming.”



Automated Standing Desk

With a Raspberry Pi for a brain, this motorised desk decides when to rise by itself. **Phil King** stands up to take a closer look



David Kong

MAKER

David describes himself as a storyteller, hacker, and optimiser. Previously in marketing at Khan Academy, he now works in product strategy and research at Frame.io.

davidkong.net

Despite having a fancy motorised sit-stand desk in his workplace, David Kong ended up sitting most of the time. “I have a lot of back and neck pain,” he confides, “and moving between sitting and standing was actually great for me, when I did it. But I just never remembered to.”

His solution? A desk that would rise automatically every hour or so, encouraging him to spend more time standing. “One of the most valuable takeaways from behavioural economics is the insight that you’re far, far more likely to do the ‘right’ thing when it’s also the easiest thing, or the default thing,” he explains.

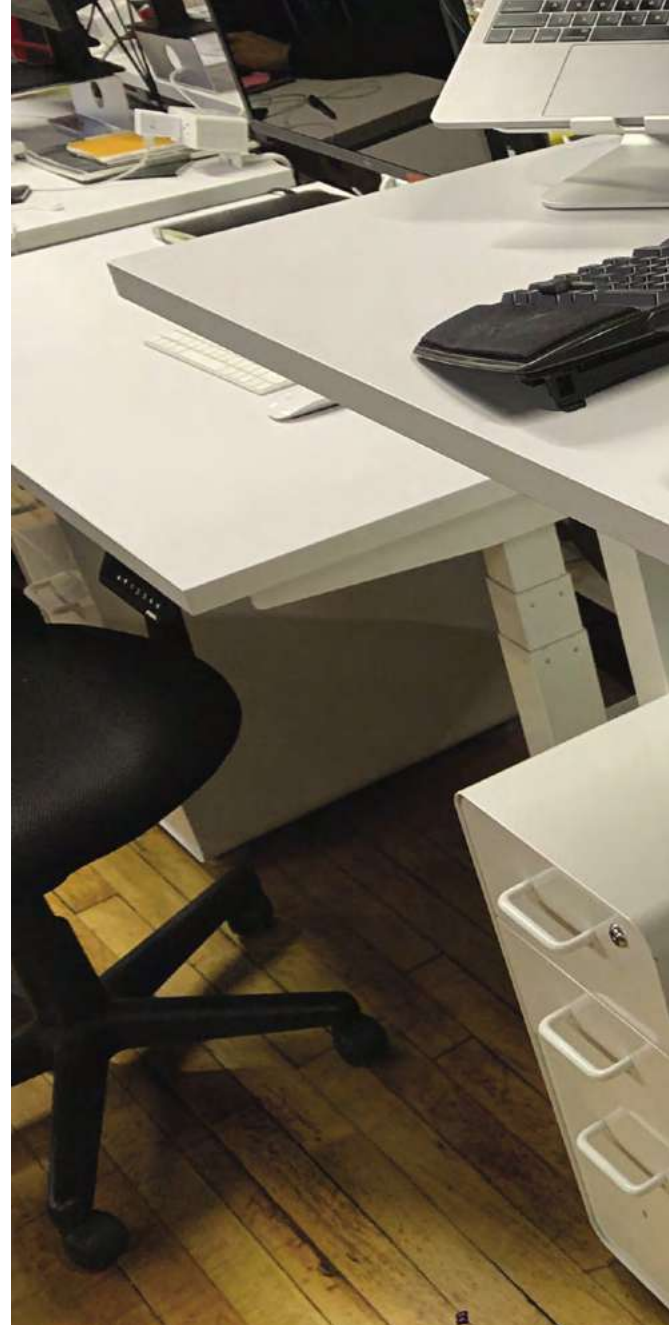
Making the connection

After opening up his desk’s control box to investigate its workings, he discovered a set of test pins that, when connected correctly, replicated the actions of the programmable buttons.

By connecting two particular pins together, he found he could cause the desk to rise to a preset height. Now all he needed was a way of connecting them in an automated way to raise it at regular intervals. The obvious choice was a relay which could be triggered with a small current. After thinking about designing a circuit with a 555 timer chip, he settled on a Raspberry Pi Zero to control the relay.

“You could argue that a Raspberry Pi was hugely overkill for this project,” he says. However, he had been wanting an excuse to play around with one for ages, and found it easy to work with. “The ability to SSH right in and to write the program in Python, which I’m much more comfortable with than C++, made it very attractive.”

The project took David a few hours to complete, altogether, over the course of a few days. “Honestly, nothing was particularly hard,” he tells



us. “I had to do a bit of research to find the right type of relay, since I’m not very familiar with all of the technical specifications for those kinds of electrical components.”

He found setting up Raspberry Pi and writing the relay control code easy. Soldering the wires – from the desk’s pins to the relay, and the latter’s control pins to Raspberry Pi’s GPIO – was also simple. “I had to shave a little bit of plastic off of the control box to allow it to close fully with my wire coming out, but that was pretty easy.”

Don’t sit down

With the desk now rising at regular intervals, David found it helped him avoid sitting for too long.

“When I had to make a conscious decision to stand, I did it far less often. The ‘default’ was to sit all the time. When the desk rose automatically every hour, the ‘default’ was to stand, and I almost always ended up standing for a good chunk of the hour.”



David's Poppin sit-stand desk could already be programmed to rise to a preset height, but he wasn't using it

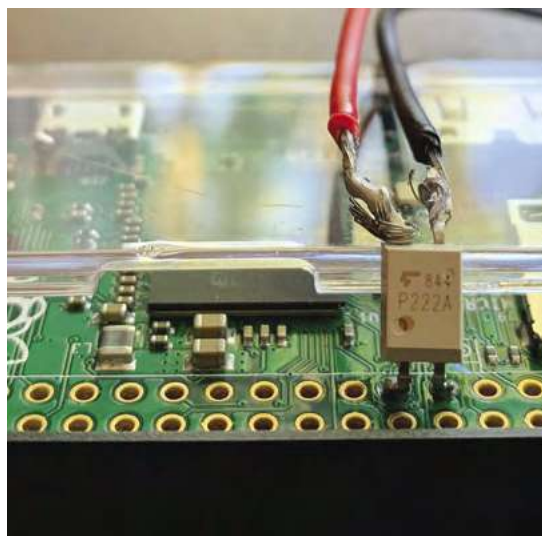
Opening up the desk's control buttons box, David discovered some test pins

By connecting a Raspberry Pi-controlled relay to the test pins, he made the desk rise automatically at regular intervals

“When I had to make a conscious decision to stand, I did it far less often”

He ended up tweaking the code to make it operate a little more randomly. “I originally had it going up at exactly the same time every hour, but then I found myself anticipating it, and I found it just more pleasant to have it happen more organically.”

While he has noticed less pain in his back as a result, the desk has occasionally taken him by surprise. “It definitely has caused some funny moments! Sometimes, I would be on a video call, in the middle of a sentence, and the desk would start to rise. I would just keep on talking and stand up with it. It never caused any problems, though, and it would sometimes be a nice way to lighten the mood.”



▲ The wires from the test pins are connected to a relay chip soldered to Raspberry Pi Zero's GND and GPIO 17 pins

Quick FACTS

- ▶ David used a four-pin Toshiba TL222A relay
- ▶ He programmed his Raspberry Pi Zero via SSH
- ▶ Project details are in his Medium piece: magpi.cc/standingdesk
- ▶ This includes the simple Python script he wrote
- ▶ By triggering the relay, it raises the desk every 45 to 60 minutes

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%



Subscriber Benefits

- ▶ **FREE Delivery**
Get it fast and for FREE
- ▶ **Exclusive Offers**
Great gifts, offers, and discounts
- ▶ **Great Savings**
Save up to 35% compared to stores

Rolling Monthly Subscription

- ▶ Low monthly cost (from £5)
- ▶ Cancel at any time
- ▶ Free delivery to your door
- ▶ Available worldwide

Subscribe for 12 Months

£55 (UK) £90 (USA)
£80 (EU) £90 (Rest of World)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero W Kit with Rolling Monthly Subscription)

📞 Subscribe by phone: **01293 312193**

📧 Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

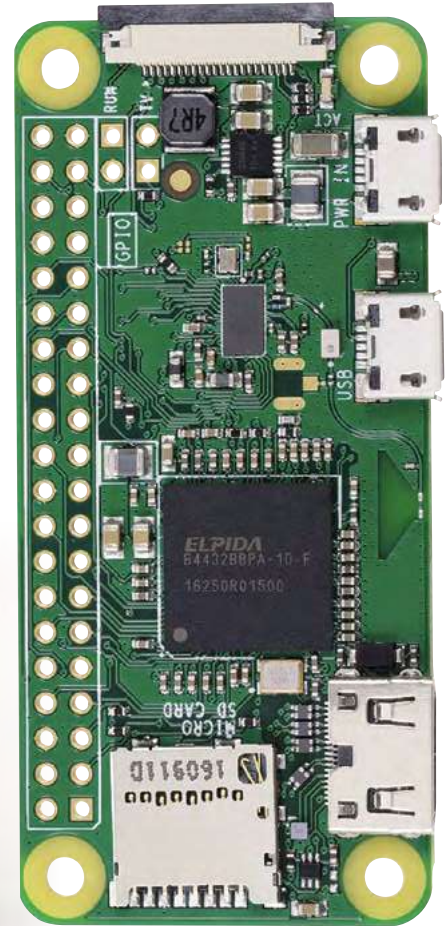
FREE Raspberry Pi Zero W Starter Kit

WITH YOUR FIRST
12-MONTH SUBSCRIPTION

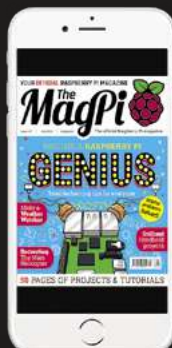
Subscribe in print
today and you'll
receive all this:

- ▶ Raspberry Pi Zero W
- ▶ Raspberry Pi Zero W case with three covers
- ▶ USB and HDMI converter cables
- ▶ Camera Module connector

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

**WORTH
£20**

Buy now: magpi.cc/subscribe



SUBSCRIBE on app stores

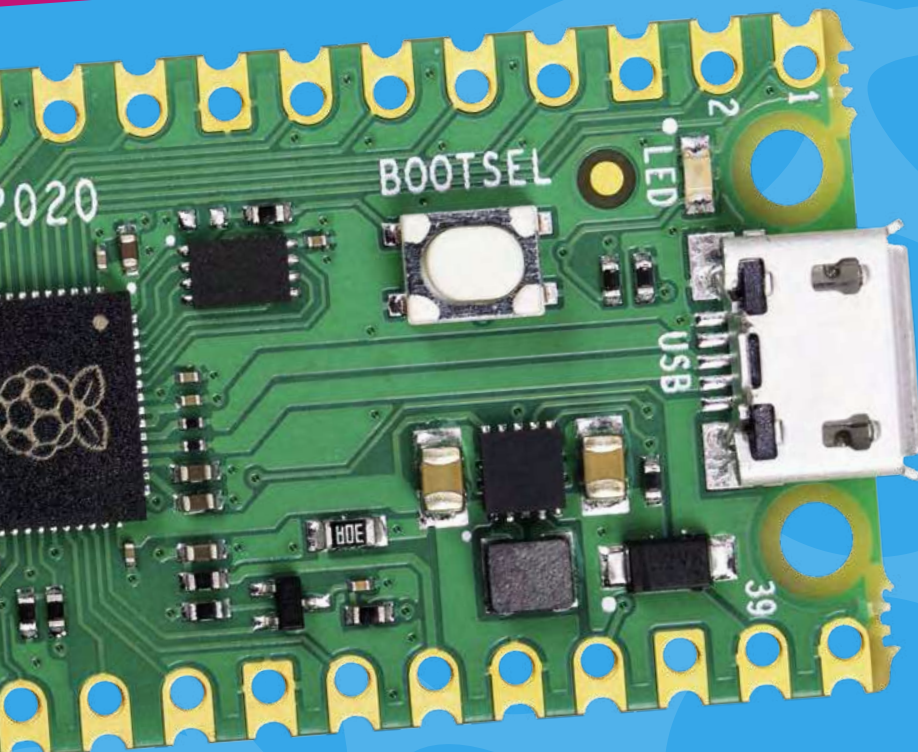
From £2.29

Available on the
App Store

GET IT ON
Google Play

20 RASPBERRY PI PICO PROJECTS

Get ideas for using your Pico with these incredible builds



It may not seem like it, but Raspberry Pi Pico came out earlier this year. It feels like it's been here longer. The \$4 microcontroller is a wonderful and deceptively powerful piece of kit that can power projects in a different way to its bigger siblings.

If you're stuck on what you want to do with a Pico, we have you covered. Over the next few pages, we're going to introduce you to some amazing things that people have made with their Pico, and hopefully spark your imagination in the process...

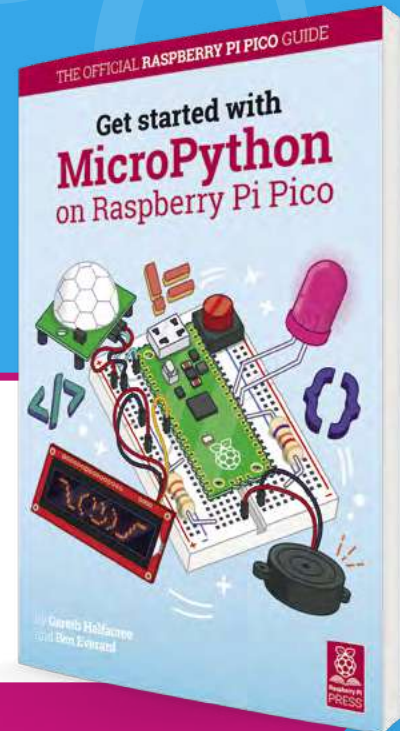
Get Started with Pico

Where to go when you're new to Pico

Get Started with MicroPython on Raspberry Pi Pico

This book (and PDF) is a great way to learn the basics of setting up your Pico, as well as giving you some basic projects to get started with so that you get some experience doing so. The appendix also has a lot of great referenced material for advanced uses.

magpi.cc/picobook



Official examples

With a load of simple examples for a Pico, this repository is a great place to get stuck in with a Pico. It also has a getting started guide that covers the use of C in Pico, which has some different functions to MicroPython.

magpi.cc/picoexamples

Raspberry Pi Pico SDK Examples

Getting started

See [Getting started with the Raspberry Pi Pico](#) and the [README](#) in the repo with the information on getting up and running.

First Examples

App	Description	Link to get it off of
Blinky	The obligatory hello world program for Pico (Output over serial)	https://github.com/raspberrypi/pico-examples
Blinky USB	The obligatory hello world program for Pico (Output over USB)	https://github.com/raspberrypi/pico-examples
Blinky	Blinky on USB instead of serial	https://github.com/raspberrypi/pico-examples

ADC

App	Description
Blinky_adc	Displays the voltage from an ADC input
Blinky_adc_gpio	Displays a digital I/O input based on the ADC inputs
Blinky_adc_gpio	An interactive shell for playing with the ADC includes examples of how to use the capture mode

Clocks

App	Description
Blinky_clock	Change the system clock frequency to 48 MHz when running
Blinky_gpio	Use the general purpose clock outputs (GPIOs) to drive pulsing of external clocks only (GPIO outputs)
Blinky_gpio	Enables the clock resourcelib feature, "resourcelib" prop the system clock and show how to use resourcelib

CMake

Full Applications

Name	Description
Blinky	This is a simple program for Pico (Output over serial)
Blinky USB	A simple hello world program for Pico (Output over USB)
Blinky	A no I/O, but functional USB sound card, hooked up via our old friend TinyUSB, USB device stack, keeping it around as it works nicely!

Audio

Name	Description
Blinky_audio	A simple hello world program for Pico (Output over USB)
Blinky_audio_gpio	A simple hello world program for Pico (Output over USB)
Blinky_audio_gpio	A simple hello world program for Pico (Output over USB)

Scanout Video

In recent video every pixel is driven by the Pico every frame and a frame buffer is not (necessarily) used (which is useful when you only have 264K of RAM).

For a full description of scanout video see [here](#)

Name	Screenshot	Description
Blinky		So named because it was the first demo program written that used video. It is a bit dated now and has a time where there was much less RAM on the Pico (264K)
Blinky_audio		Streams video data out of frame fast enough to drive 640x480x60fps on a display

Pico Playground

If you're finished with the simpler stuff and want something with a bit more bite, these projects on Pico Playground are a bit more advanced and worth a look so you can appreciate just how much a Pico can do.

magpi.cc/picoplayground

Gaming with Pico

Play games or emulate them on a tiny microcontroller

PICO GAME BOY

LyneByLyne

magpi.cc/picogameboy

This neat hack of a Game Boy Color allows you to run homebrew Game Boy ROMs from a Pico, complete with a menu to select the game. A bit like those old 52 games-in-one, but the games are actually good. You're also getting the true play experience on original hardware instead of emulation. This makes the idea of keeping a Game Boy in your bag or pocket much more desirable, especially with the plan for loading from microSD.



1.

PICO PONG

Nick Bild

magpi.cc/picopong

We interviewed Nick about this gesture-controlled Pong project in issue 107 (magpi.cc/107), which involved him first getting all the hardware to work together over the course of a few weeks, before finally coding his own version of Pong to work with it in a couple of hours. Why gesture control? "I wanted to do something different from the typical potentiometer controller."



3.

2.



T-REX GAME CONTROLLER

NarayanDojo

magpi.cc/trexgun

Due to the way CRT and modern TVs differ in the way they draw images on a screen, old light gun games can no longer work. While this is not a full conversion of an old light gun into a working modern light gun, it does allow for it to become a macro button. Merge it with some motion/gesture control and you basically have a modern light gun setup.

4.



CUSTOM USB GAMES CONTROLLER

Ben Everard

magpi.cc/picopad

In the same way a Pico can be used to make a custom keyboard, you can also use it to make custom controllers – although the project here makes greater use of the keyboard input stuff. Still, Ben from HackSpace magazine walks you through creating your own custom joystick which can easily be modified and customised to your own preferences. All the inputs can be connected directly to a Pico's GPIO as well.

“ This extremely small console plays custom-made games based on Snake and Tetris ”



BOP IT MINECRAFT CONTROLLER

Seth Altobelli

magpi.cc/bopitcraft

We talked to Seth last issue about this (magpi.cc/108). In the video introducing the concept, he jokes about Minecraft being too easy and wanting a way to challenge himself, resulting in a custom controller made from a Bop It. It works by using the original buttons and switches and such, which were able to be soldered directly to Pico's GPIO inputs. There's also an accelerometer so the mouse can be moved by the Bop It in the air.

TINY GAME CONSOLE

Theking3737

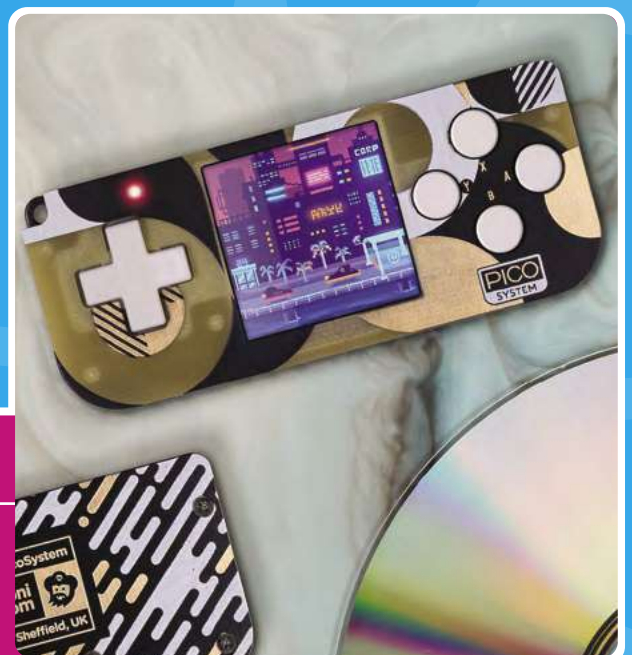
magpi.cc/tinyconsole

Pico can display video, and it can take inputs, which can mean only one thing: a handheld games console was bound to happen. With a cool 3D-printed case, this extremely small console plays custom-made games based on Snake and Tetris, and can be used both vertically and horizontally. It's pretty cute.



PICOSYSTEM

Using an RP2040, the same chip as in Raspberry Pi Pico, Pimoroni has turned a development kit into a games console: magpi.cc/picosystem



Cool and practical

Great projects that make use of Pico power

MANDELBROT SET

iridis-alpha

magpi.cc/picofract

It may not look like it at first glance, but Raspberry Pi Pico can actually render video if you program it the right way, and have an appropriate display. Using it you can create cool, animated images from the Mandelbrot set – a set of complex numbers that creates a hypnotic graph for complicated maths reasons. Picofract is a bit of code that allows you to do this and makes use of both cores of Pico.



7.

MATRIX TOUCHSCREEN KEYBOARD

Dustin Watts

magpi.cc/matrixtouch

Taking the ability to act as a keyboard and translating it to a touchscreen is a great way to make a more custom interface for your own use, a public access point, or even in a game that interacts with the real world. This project uses a generic touchscreen with a custom PCB to connect it to Pico that you can buy from Dustin.



8.

MACRO KEYPAD

BlobbyBlue02

magpi.cc/keypad

As we mentioned with the mechanical keyboard, you can easily create a customisable keyboard, keypad, macro buttons, or stream deck with Pico and some keys. With MicroPython, you can pass a combination of keys at once with a press of a button, so creating macros such as undo, paste, save and exit, open vim, etc., are fully possible with the right amount of tweaking. We'd like to know what the Mario button does, though.



10.

MECHANICAL KEYBOARD

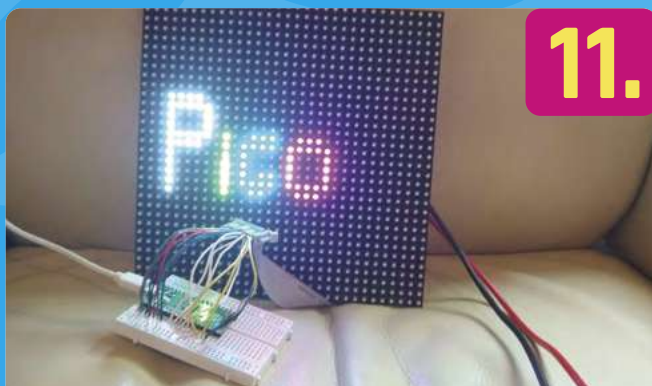
Adam K

magpi.cc/envkb

One of the cool functions of Pico is its ability to act as a human input device, such as a keyboard. This is completely hackable by you, which means you can create any keyboard layout you wish... as well as a standard one, like this excellent Env-KB mechanical keyboard, which is one part code and one part excellent PCB design. The maker plans to create the PCB with a dedicated RP2040 in the future.



9.

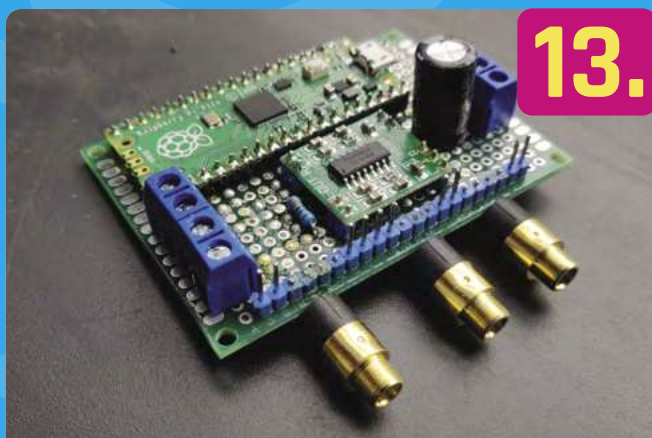


11.

ANIMATED SIGN

Andrew Gregory
magpi.cc/picosign

This project from our sibling publication, HackSpace magazine, uses a HUB75 LED panel, a sizeable display that creates amazing multicoloured images and can even be used as part of a bigger display with multiple panels. With so many LEDs (1024) to address, the code can get a little long unless you employ some clever loops.

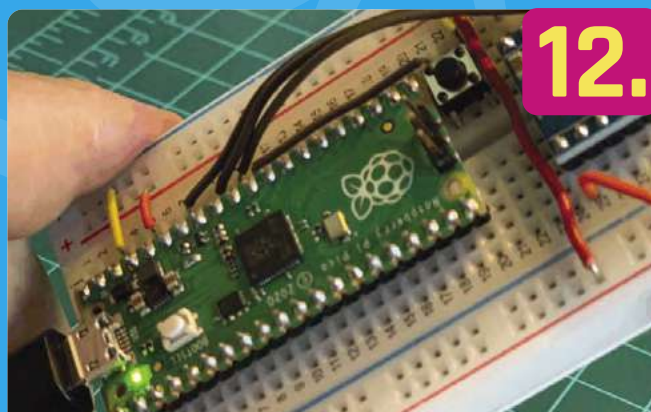


13.

SPEAKER AND SENSOR READER

pubudex
magpi.cc/speaksense

This is a custom prototype board for connecting to an exercise bike, and is able to read all the various sensors on the bike, as well as amplifying any music connected to it. Sure, the exercise bike can probably tell you some of the data, but it's much more fun to record it and display it in a way you prefer for better analysis later.

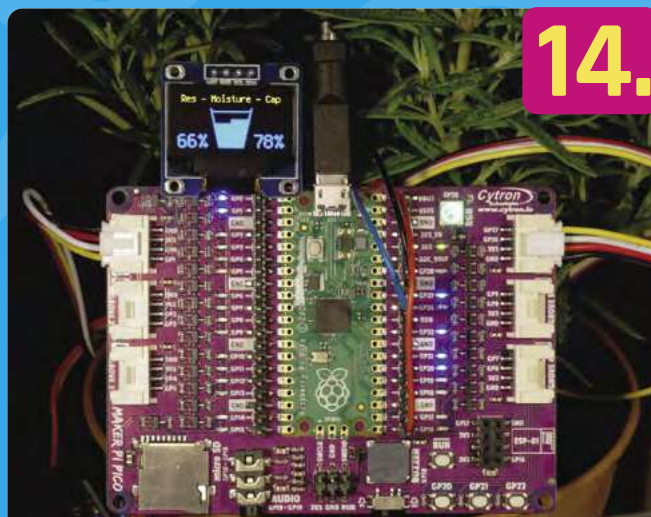


12.

TINYML

Marcelo Rovai
magpi.cc/tinymml

This build doesn't look very fancy, but it really is – Marcelo has added gesture recognition and machine learning to it. As well as being a great exploration of the abilities of Pico, it's an interesting way to teach a model about how people actually perform gestures with their arms. He's also managed to use it to detect the contents of a blender to make sure you're making a drink correctly.



14.

SOIL MOISTURE SENSING

Kevin Walters
magpi.cc/picosoil

Garden automation is something we like over here in MagPi Towers, and seeing someone like Kevin make great use of Cytron's Maker Pi Pico to do some basic soil moisture sensing is great. This project also uses the optional OLED screen and displays the resistance of the soil, as well as the capacitance across two points to create the readings on it.

Musical Pico

Create cool music with these custom instruments



TRILL GUITAR

Mike Cook
magpi.cc/trillguitar

15.

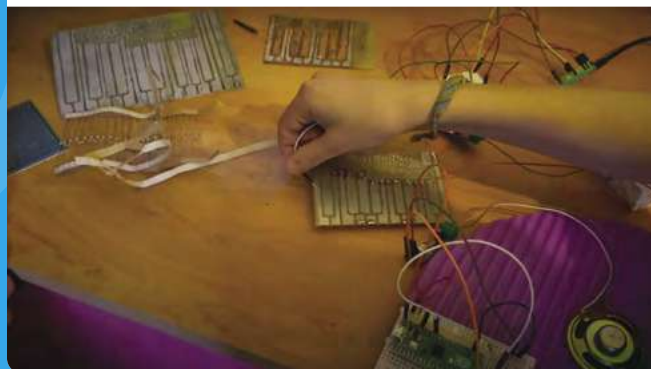
This MIDI guitar created by *The MagPi* regular Mike Cook uses Trill touch sensors, a Pico, and a custom-made case to simulate playing a real guitar. You can follow along to the tutorial series starting in issue 102 of *The MagPi* (magpi.cc/102). A full-size Raspberry Pi handles the synth part, while Pico monitors and displays info on the 'head' of the guitar. Pressing frets will change the chord, which is displayed via Pico.

PICO PIANO

Amelie, Caleb, and Ziva
magpi.cc/picopiano

16.

A very DIY piano created for the Raspberry Pi Foundation's Digital Making at Home livestream (rpf.io/home) includes copper-plated boards and a spare pin to connect to the notes to create a circuit and play chords. It's all connected up to a Pico and while you can't quite tickle the ivories, with some perseverance you can make some great music.



MIDI FIGHTER

Liz Clark
magpi.cc/midifighter

17.

These types of controllers are called MIDI Fighters due to the way they use a grid of arcade machine buttons. No Street Fighter-style 'Dragon Punch' (DP) motions are required to play it: just hammer the buttons linked to your samples or notes to make great music. It uses an expander board to connect all the arcade buttons to Pico, and includes an LED screen to show you info on the current chords.



PORTAL MUSIC BOX

thisoldgeek
magpi.cc/picomusicbox

18.

This fun little project plays a MIDI version of Still Alive from the Portal video game on an appropriately themed, 3D-printed box. It uses a mixture of Pico and Adafruit speakers to play the tinny sound after a small conversion through musicboxmaniacs.com. The print of the box is the longest part of the project at about six hours.



Robot Pico

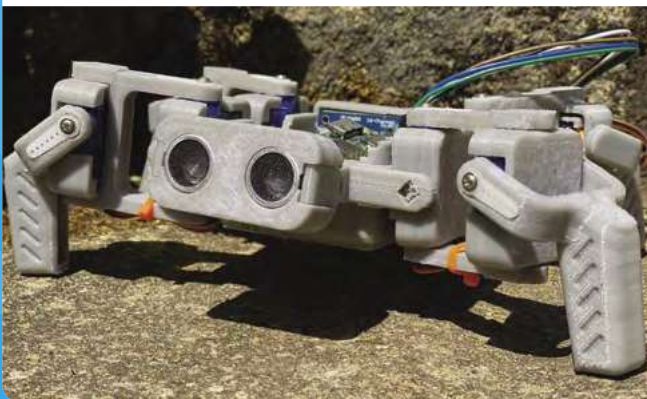
Upgrade your automatons with some Pico sensors

PICO SMARS

Kevin McAleer
magpi.cc/picosmars

19.

SMARS (Screwless Modular Assemblable Robotic System) is a 3D-printable robot created for education. As it's very modular and customisable, using Pico to control motors and monitor sensors is an obvious choice. Pico interfaces with a motor board for the movement part, but is able to natively read the ultrasonic distance sensor. The video goes through how you can make your own Pico SMARS.



PICO LINE FOLLOWER ROBOT

RoboCircuits
magpi.cc/picolinebot

20.

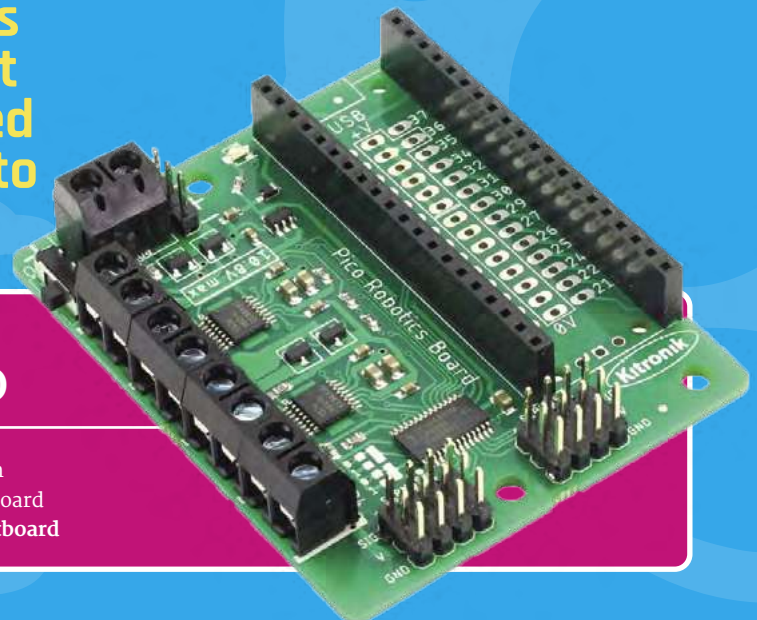
Line following is a very simple task that a lot of hobby robots can undertake. This robot uses a custom PCB with a Pico and robot parts mounted to it, including a line follower – a kind of IR sensor that reacts differently to black surfaces. Using a loop, a robot then tries to keep on the black line. This is also a competition you can find in Pi Wars.



“No ‘Dragon Punch’ motions are required to play it: just hammer the buttons linked to your samples or notes to make great music”

ROBOTICS BOARD FOR PICO

Don't fancy creating your own custom robotics PCB? The Kitronik Robotics Board might be for you: magpi.cc/picorobotboard

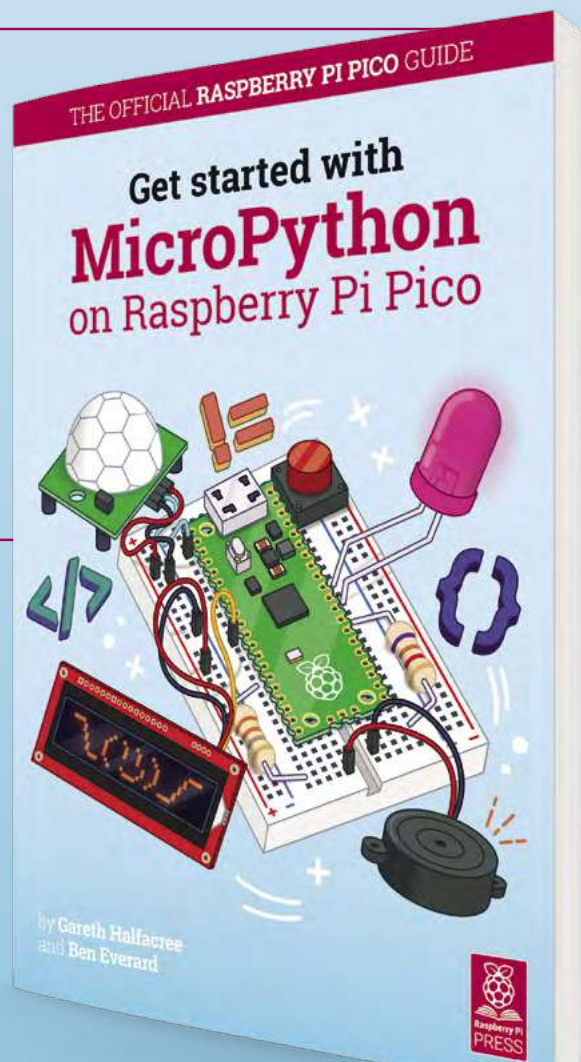




Get started with MicroPython on Raspberry Pi Pico

Learn how to use your new Raspberry Pi Pico microcontroller board and program it using MicroPython. Connect hardware to make your Pico interact with the world around it. Create your own electro-mechanical projects, whether for fun or to make your life easier.

- Set up your Raspberry Pi Pico and start using it
- Start writing programs using MicroPython
- Control and sense electronic components
- Discover how to use Pico's unique Programmable IO



Available now: magpi.cc/picobook

Ultimate home server: sharing and storing files



PJ Evans

MAKER

PJ is a writer, software engineer and tinkerer. His server needs to go on a diet.

twitter.com/mrpjevans

Sharing and storing large files on your network equals lots of fun and productivity without filling up your computer

Have you got that file? Can you email it to me?" "But it's 10GB! It's too big!" We've all been there. If you're in a house full of data-hungry family members, or working in a small business where a single repository of information would be enormously helpful, then a file-sharing device (sometimes called 'network-attached storage' or NAS) is what you need. We're going to take the Argon M.2 Raspberry Pi server we built last month and add file sharing and increase disk capacity so there's plenty of room for all your stuff.

01 Choose the right model

To create our ultimate file sharing device, we need to install some file-sharing software. There are many different protocols and options

▼ Identify your drive carefully before removing and creating partitions

```

Disk /dev/sda: 931.5 GiB, 1000204886016 bytes, 1953525168 sectors
Disk model: Forty
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 33553920 bytes
Disklabel type: dos
Disk identifier: 0x7686e0f2

Device      Boot  Start        End      Sectors  Size Id Type
/dev/sda1                8192     532479      524288  256M c W95 FAT
/dev/sda2          532480 1953525167 1952992688  931.3G 83 Linux

Disk /dev/sdb: 3.7 TiB, 4000752599040 bytes, 7813969920 sectors
Disk model: Elements 2621
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 7AD6A69C-7694-9348-82A7-2192E4FE86A9

Device      Start        End      Sectors  Size Type
/dev/sdb1    2048 7813969886 7813967839  3.7T Linux filesystem
pj@ultimate:~$ _

```

available to us. It makes sense to choose a protocol that is supported by as many different platforms as possible, such as Linux, Windows, and macOS. The de facto standard is SMB (Server Message Block) and the software that implements SMB on Raspberry Pi OS is called Samba. Installation is simple:

```
sudo apt -y update && sudo apt -y upgrade
sudo apt install samba
```

If you get asked a question about WINS, just say no. After installation completes, you have a running file server, but with nothing configured.

02 Prepare your sharing location

The next choice to make is where to store the files that we wish to share on the network. There's no hard and fast rule on this. You can use your user's home directory or create a dedicated area. We're going to create a new directory under `/var`, which is an area in the Linux file system designated for files that will 'vary'.

```
sudo mkdir /var/shared
```

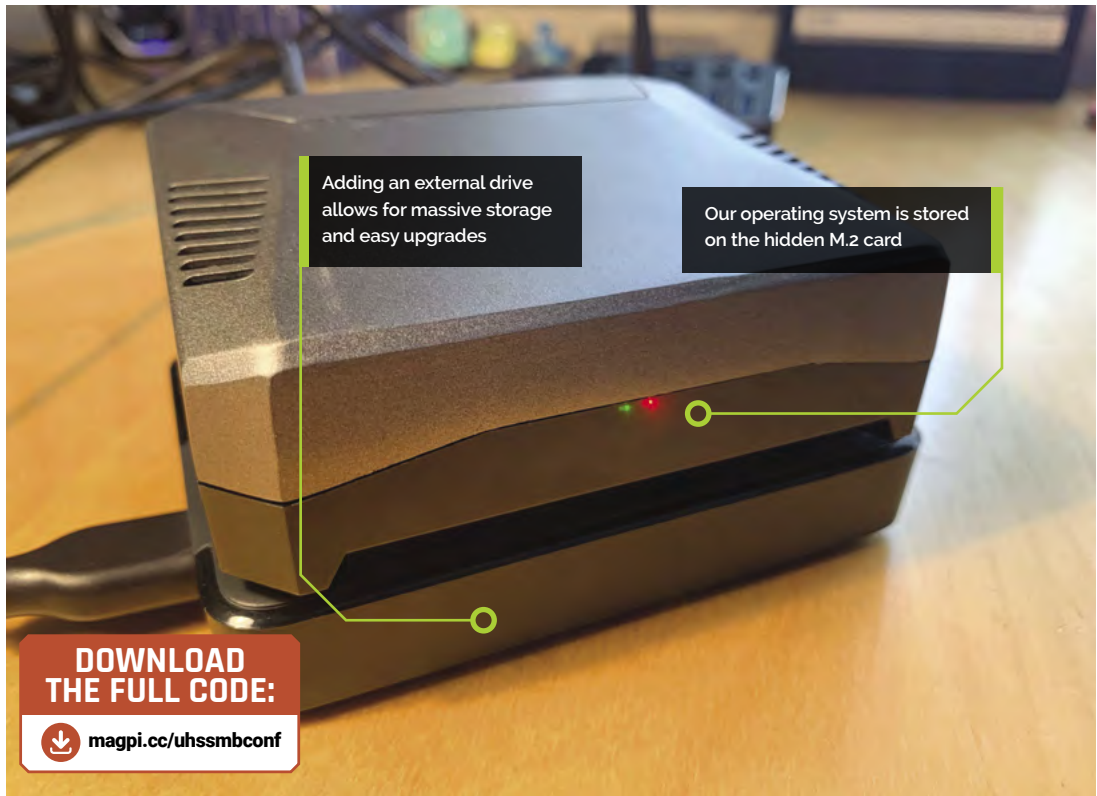
We're now going to allow any user to read and write files in that area:

```
sudo chmod -R 777 /var/shared
```

Putting the 'welcome mat' down like this is not considered best practice, but it serves our purposes. For critical implementations, investigate more granular read/write settings.

You'll Need

- Raspberry Pi ultimate server magpi.cc/108
- 4TB Western Digital USB 3.0 external drive (or similar) magpi.cc/wdelements
- Powered USB hub magpi.cc/usbhub



03 Configure Samba

We need to tell Samba to share our new directory. All the sharing settings and configuration options are stored in the file `/etc/samba/smb.conf`, so open that up in an editor:

```
sudo nano /etc/samba/smb.conf
```

This is a large file, but don't be put off: we're just going to add to the end of it. Add the text shown in the code listing for `smb.conf`.

Take care to indent the code correctly. These lines set up the most basic kind of sharing. Save the file and exit the editor (**CTRL+X**), then restart Samba so the new configuration is picked up.

```
sudo systemctl restart smbd
```

smb.conf

➤ Language: **Config / Bash**

```
001. [Shared]
002.     comment = Shared
003.     path = /var/shared
004.     read only = no
005.     browsable = yes
```

Getting connected

How you connect to remote servers varies by platform, but we'll cover some common ones here.

- **Raspberry Pi Desktop:** Open the File Manager, click 'Go' in the menu bar, click 'Network', select 'ULTIMATE' (or whatever you named your server), click the 'Shared' directory, and enter your credentials (leave the domain as WORKGROUP).
- **macOS:** Open Finder, click on 'Network' on the sidebar, select 'ULTIMATE' (or whatever you named your server), click 'Connect As...', then enter your username and Samba password. Click 'Shared'.
- **Windows:** Open File Explorer and right-click on 'This PC', then select 'Add a network location'. You'll go through a short wizard; when asked which server to connect to, enter `\\ULTIMATE\\Shared` and enter your username and Samba password.

04 Configure your users

Security constraints mean Samba cannot check your password when you connect. To mitigate this, you can set an additional password for Samba access. This can be the same as your regular password for something different. To add your user account to Samba and set your access password, just run the following command:

```
sudo smbpasswd -a <user>
```


...Where `<user>` is your current username or any other username that needs Samba permissions. Enter your new password twice and you're ready to go.

05 Get yourself connected

Samba is now ready for use. You can also mount shared drives from the command line in Raspberry Pi OS using `smbmount`, or connect to your Raspberry Pi from another computer on your network – see 'Getting connected'.

06 Adding more storage

Now you have a shared drive up and running, you may want to load it up with videos, audio, and more. Soon you may need more space than our M.2 drive provides. You may also wish to consider separating data and the OS onto different disks so it's easier and safer to perform upgrades or expansions. Thankfully, the powerful USB 3.0 ports on Raspberry Pi 4 are perfect for adding inexpensive larger external drives, such as the whopping 4TB Western Digital drive we've chosen. If you want to go for multiple drives, we recommend a powered USB hub to make sure there are no 'brown-outs' that could cause lost data.

07 Identify the new drive

When you plug in a USB 3.0 external drive, Raspberry Pi OS will recognise it and add it as a new device in the `/dev` directory. We need to find the assigned name of the device. Typically it will be `/dev/sdb`, but not always. We can check with this command:

```
sudo fdisk -l
```

The output will show two physical devices, labelled as 'Disks': your M.2 drive and the new external drive. It should be easy to tell them apart by the size and manufacturer name. Look for the 'logical name' of the external drive and make a note of it.

08 Partition the disk

To create a suitable file system, we first need to partition the new disk. Run the following

command (assuming your external drive's logical name is `/dev/sdb`):

```
sudo fdisk /dev/sdb
```

FDISK is a utility for managing your disk's partitions – the areas where data can be stored.

Now complete the following steps (note that this will erase any data on the disk). Repeatedly enter 'd' followed by **RETURN** to delete any current partitions until you get a 'no partition' error. Enter 'n' followed by **RETURN** to create the new partition, accepting all the defaults. Enter 'w' to write the changes to the disk and exit

09 Format the disk

Once a disk has been partitioned, it needs a file system. There are many to choose from, but a popular choice in the Linux world (and therefore Raspberry Pi OS) is ext4. To start, we need the partition name. If you have a look in the device directory (`ls /dev`), you will see that `sdb` (our drive) has been joined by `sdb1` (our partition). To format the partition:

```
sudo mkfs.ext4 /dev/sdb1
```

This will probably take a few minutes to run based on the size of the disk.

10 Mount the disk

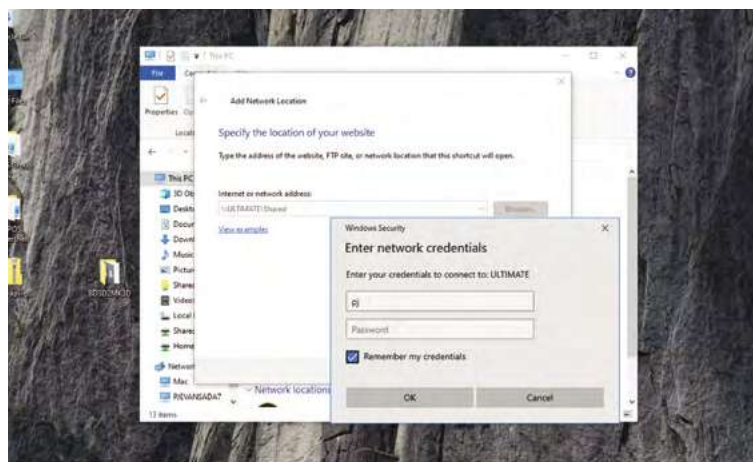
To access your fancy new drive, you need to 'mount' it. This is Linux-speak for making the formatted partition available as a directory on the

Top Tip

RAID

Did you know you can use multiple disks to provide protection against drive failure? RAID is a system for keeping two or more drives in constant sync, so if one fails, the other has your data!

▼ Windows supports connections too, but it's a little trickier





▲ Our server connected to an external drive

Top Tip

Data danger!

Using FDISK and MKFS can cause data loss. Check all your device logical names are correct and that there's no data on the disk that you need to keep.

system. The first step is to create an 'endpoint': a folder that will act as the gateway to the new disk. The normal directory to use for this is 'mnt'.

```
sudo mkdir /mnt/huge
```

You don't have to call it 'huge' – use any name you like. You only have to do this once; the folder will persist like any other.

To mount the drive:

```
sudo mount -t ext4 /dev/sdb1 /mnt/huge
```

That's it. Any files written into that folder will be placed on our new disk. To unmount:

```
sudo umount /mnt/huge
```

11 Automount on boot

The problem with the mount command is that it will not persist after a reboot. You can add a mount command to something like **rc.local**, but the best way to deal with this is to add the drive to the file system table (fstab). This is a dangerous step as a misconfigured fstab can make your Raspberry Pi unbootable. First, get the UUID of your new drive. Run this command:

```
sudo blkid
```

Find the line containing our new drive (**/dev/sdb**) and copy the list of numbers and letters quoted in **UUID=**". Now edit the table:

smb.conf (part 2)

> Language: **Config / Bash**

```
001. [Huge]
002.     comment = Huge
003.     path = /mnt/huge
004.     read only = no
005.     browsable = yes
```

```
sudo nano /etc/fstab
```

Carefully append the following line:

```
UUID=c13507ef-190d-400d-9fc0-96f7cedea521
/mnt/huge ext4 rw,user,auto 0 0
```

This is an example UUID: Make sure everything after **UUID=** is the UUID of your new drive. Save and reboot. Your drive should be automatically mounted.

“ To access your fancy new drive, you need to 'mount' it ”

12 Share the disk

The final step is to let Samba know about our new disk. Set the permissions and edit the Samba config file as we did earlier:

```
sudo chmod -R 777 /mnt/huge
sudo nano /etc/samba/smb.conf
```

At the bottom of the file, add a new line and add the contents of the **smb.conf (part 2)** listing. Save the file and exit (**CTRL+X**). You can now restart Samba to start the share without rebooting:

```
sudo systemctl restart smbd
```

Now try mapping from a remote device. All being well, you now have a huge amount of storage at your disposal.

Next time

We've now got our server sharing files across your network. So, what else can you do with an always-on device? Next month we'll look at one of the most popular applications: media streaming. 📺

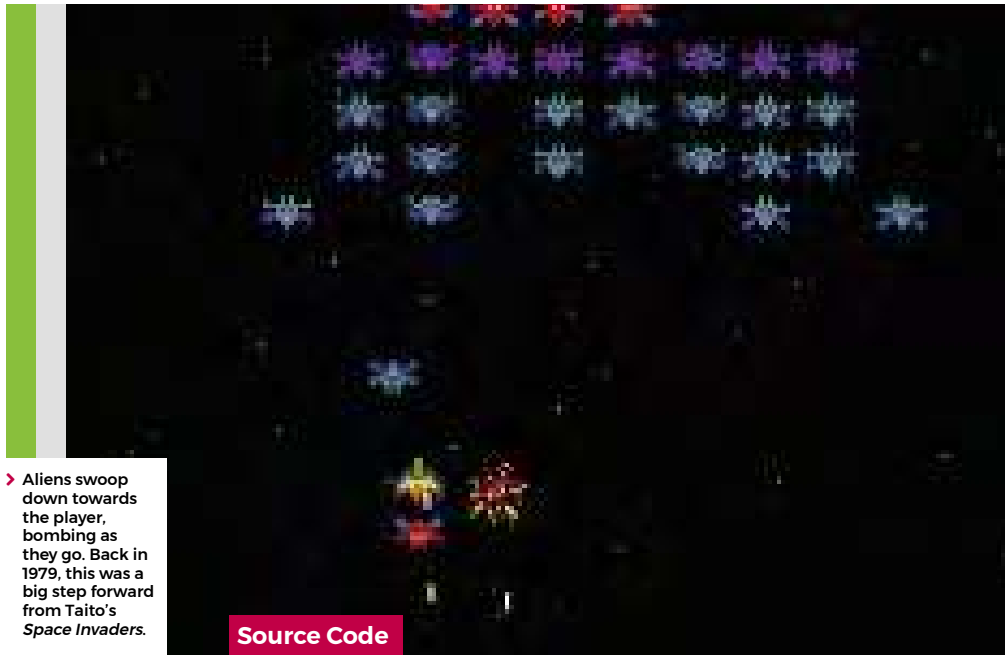
HIGHPI PRO

• ————— The new case from the HiPi.io team ————— •



- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:



➤ Aliens swoop down towards the player, bombing as they go. Back in 1979, this was a big step forward from Taito's *Space Invaders*.

Source Code



Wireframe

This tutorial first appeared in Wireframe, our sister magazine that lifts the lid on the world of video games. Every issue includes tutorials and in-depth interviews, along with news and reviews of the latest indie and triple-A games.

To find out more, visit their website at wfmag.cc.

Check out their subscription offers at wfmag.cc/subscribe.



AUTHOR
MARK VANSTONE

Recreate Galaxian's iconic attack patterns

Blast dive-bombing aliens in our salute to Namco's classic

Hot on the heels of the original *Space Invaders*, *Galaxian* emerged as a rival space shooter in 1979. Released by Namco, *Galaxian* brought new colour and unpredictable motion to the alien enemy, who would swoop down on the defending player. *Galaxian* was so popular in arcades that Namco released a sequel, *Galaga*, two years later – that game complicated the attack patterns even more. It's difficult to say how many ports and clones have been made of *Galaxian*, as there are several versions of similar games for almost every home platform.

The player's role in *Galaxian* is similar to *Space Invaders*, in that they pilot a ship and need to destroy a fleet of aliens. With *Galaxian*, however, the aliens have a habit of breaking formation and swooping down towards the player's ship, and dive-bombing it. The aim is to destroy all the enemy ships and move on to the next wave. The subsequent waves of enemies get more difficult as the player progresses. For this sample, we're going to look at that swooping mechanic, and make the bare nuts and bolts of a *Galaxian* game with Pygame Zero.

First, *Galaxian* has a portrait display, so we can set the play area's width and height to be 600 and 800 respectively. Next, we can create a scrolling backdrop of stars using a bitmap that we blit to the screen and move downwards every update. We need a second blit of the stars to fill in the space that the first one leaves as it scrolls down, and we could also have another static background image behind them, which will provide a sense of depth.

Next, we set up the player ship as an Actor, and we'll capture the left and right arrow keys in the `update()` function to move the ship left and right on the screen. We can also fire off a bullet with the **SPACE** bar, which will travel up the screen until it hits an alien or goes off the top of the screen. As in the original *Galaxian*, you can only shoot one bullet at a time, so we only need one Actor for this.

The aliens are arranged in rows and move left and right across the screen together. We'll stick to just one type of alien for this sample, but draw two rows of them. You could add extra types and any number of rows. When we create the alien Actors, we can also add a status flag, and we need to determine which side of the row they're

on as when they break formation, the two sides fly in opposite directions. In this case, there'll be four aliens on the left of each row and four on the right. Once they're set up in a list, we can iterate through the list on each update and move them backwards and forwards. While we're moving our aliens, we can also check to see if they've collided with a bullet or the player ship. If the collision is with a bullet, the alien cycles through a few frames of an explosion using the status flag, and then, when their status reaches five, they're no longer drawn. If the collision is with the player, then the player dies and the game's over. We can also check a random number to see if the alien will start a bombing run; if so, we set the status to one, which will start calls to the `flyAlien()` function. This function checks which side the alien's on and starts changing the alien's angle, depending on the side. It also alters the x and y co-ordinates, depending on the angle. We've written this section in longhand for clarity, but this could be collapsed down a bit with the use of some multiplier variables for the x co-ordinates and the angles.

There we have it: the basics of *Galaxian*. Can you flesh it out into a full game? 🐘



Download
the code
from GitHub:
[wfmag.cc/
wfmag50](https://wfmag.cc/wfmag50)

Massive attack

Here's Mark's dive-bombing *Galaxian* code. To get it working on your system, you'll need to install Pygame Zero – full instructions are available at wfmag.cc/pgzero.

```
# Galaxian
from random import randint
WIDTH = 600
HEIGHT = 800

bullet = Actor('bullet', center=(0, -10))
ship = Actor('ship', center=(300, 700))
backY = count = gameover = 0
aliens = []

for a in range(0, 8):
    aliens.append(Actor('alien0', center=(200+(a*50),200)))
    aliens[a].status = 0
    aliens[a].side = int(a/4)

for a in range(0, 8):
    aliens.append(Actor('alien0', center=(200+(a*50),250)))
    aliens[a+8].status = 0
    aliens[a+8].side = int(a/4)

def draw():
    screen.blit("background", (0, 0))
    screen.blit("stars", (0, backY))
    screen.blit("stars", (0, backY-800))
    bullet.draw()
    drawAliens()
    if gameover != 1 or (gameover == 1 and count%2 == 0): ship.
draw()

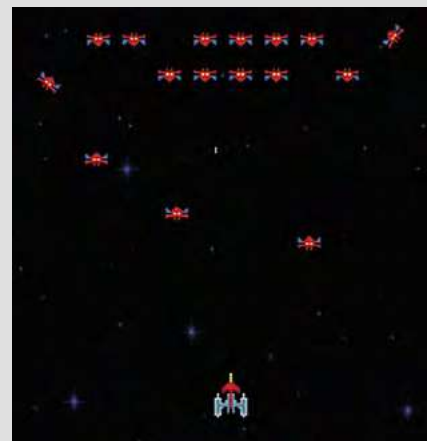
def update():
    global backY, count
    count += 1
    if gameover == 0:
        backY += 0.2
        if backY > 800: backY = 0
        if bullet.y > -10: bullet.y -= 5
        if keyboard.left and ship.x > 50 : ship.x -= 4
        if keyboard.right and ship.x < 550 : ship.x += 4
        if keyboard.space :
            if bullet.y < 0: bullet.pos = (ship.x,700)
            updateAliens()

def drawAliens():
    for a in range(0, 16):
        if aliens[a].status < 5 : aliens[a].draw();

def updateAliens():
    global gameover
    for a in range(0, 16):
        aliens[a].image = "alien0"
        if count%30 < 15 : aliens[a].image = "alien1"
        if count%750 < 375:
            aliens[a].x -=0.4
```

```
else:
    aliens[a].x +=0.4
    if aliens[a].collidepoint(bullet.pos) and aliens[a].
status < 2:
    aliens[a].status = 2
    bullet.y = -10
    if aliens[a].colliderect(ship) : gameover = 1
    if randint(0,1000) == 1 and aliens[a].status == 0 :
aliens[a].status = 1
    if aliens[a].status == 1 : flyAlien(a)
    if aliens[a].status > 1 and aliens[a].status < 5:
        aliens[a].image = "alien" + str(aliens[a].status)
        aliens[a].status += 1

def flyAlien(a):
    if aliens[a].side == 0:
        if aliens[a].angle < 180 :
            aliens[a].angle += 2
            aliens[a].x -= 1
            if aliens[a].angle < 90: aliens[a].y -= 1
        if aliens[a].angle >= 90 :
            aliens[a].y += 2
        if aliens[a].angle >= 180 :
            aliens[a].angle = 180
            aliens[a].x += 1
    else:
        if aliens[a].angle > -180 :
            aliens[a].angle -= 2
            aliens[a].x += 1
            if aliens[a].angle > -90: aliens[a].y -= 1
        if aliens[a].angle <= -90 :
            aliens[a].y += 2
        if aliens[a].angle <= -180 :
            aliens[a].angle = -180
            aliens[a].x -= 1
```



◀ Our homage to the classic *Galaxian*, with angry aliens that love to break formation.

sinclair



Commodore

the

COMPUTERS

THAT MADE

BRITAIN



OUT
NOW

***"The Computers that Made Britain**
is one of the best things I've read
this year. It's an incredible story of
eccentrics and oddballs, geniuses and
madmen, and one that will have you
pining for a future that could have been.
It's utterly astonishing!"*

- **Stuart Turton**, bestselling author
and journalist

.....



Buy online: wfmag.cc/ctmb

Available on
amazon

Create GUIs with Python: Paint

Create a fun picture-matching game



Laura Sach

Laura leads the A Level team at the Raspberry Pi Foundation, creating resources for students to learn about Computer Science.

@CodeBoom



Martin O'Hanlon

Martin works in the learning team at the Raspberry Pi Foundation, where he creates online courses, projects, and learning resources.

@martinohanlon

You are going to build a simple application which will allow you to paint using lines and shapes (Figure 1). You will create your paint application in four stages:

- drawing dots which follow the mouse
- draw lines between the dots
- adding colours and line width modifier
- drawing shapes

Note that you can style your application anyway you want – it doesn't have to look like this one.

Drawing dots

The first step is to create a simple application which will use the Drawing widget and the `when_mouse_dragged` event to draw dots (or ovals on the screen). In the imports section of your otherwise blank program, add the widgets:

```
from guizero import App, Drawing
```

Create a new function:

```
def draw(event):
    painting.oval(
        event.x - 1, event.y - 1,
        event.x + 1, event.y + 1,
        color="black")
```

Add this code to the app section:

```
app = App("Paint")

painting = Drawing(app, width="fill",
height="fill")

painting.when_mouse_dragged = draw
```



▲ **Figure 1** The finished game

```
app.display()
```

Your code should resemble **paint1.py** (overleaf). The Drawing widget fills all the available space on the window. When the mouse is dragged across the drawing, the function **draw** is called, which draws ovals on the painting.

The **draw** function is called each time an event is raised. The event which contains the x and y position of the mouse is passed as a variable to the function.

There is a problem, though. Unless you move your mouse very slowly, a series of dots is drawn by your program, not a continuous line

paint1.py

> Language: **Python 3**

**DOWNLOAD
THE FULL CODE:**

 magpi.cc/guizero

```
001. # simple paint app, just draw dots
002.
003. # -----
004. # Imports
005. # -----
006.
007. from guizero import App, Drawing
008.
009. # -----
010. # Functions
011. # -----
012.
013. def draw(event):
014.
015.     painting.oval(
016.         event.x - 1, event.y - 1,
017.         event.x + 1, event.y + 1,
018.         color="black")
019.
020. # -----
021. # App
022. # -----
023.
024. app = App("Paint")
025.
026. painting = Drawing(app, width="fill", height="fill")
027.
028. painting.when_mouse_dragged = draw
029.
030. app.display()
```

(Figure 2). It's not a very good paintbrush! There are gaps between the dots because an event is not raised for every pixel the mouse crosses.

Lines between the dots

To solve this problem, you are going to change the program to draw lines between the dots. That way, the line made will be continuous and be more like a pen or paintbrush.

You will need to use a `when_left_button_pressed` event to store the position of where the line starts. Then draw a straight line between where the line starts and the next position the mouse was dragged to. Create a new function which will be called when the mouse is pressed:

“ You are going to change the program to draw lines between the dots ”

```
def start(event):
    painting.last_event = event
```

Add this to the app section:

```
painting.when_left_button_pressed =
start
```

Now, add the Picture widget to your imports:

```
from guizero import App, Box, Picture
```

The position of where the line starts is stored in the `last_event` variable.

Modify the `draw` function to draw a line between where the line starts and where the mouse has been dragged to.

```
def draw(event):
    painting.line(
        painting.last_event.x, painting.
last_event.y,
        event.x, event.y,
        color="black",
        width=3
    )

    painting.last_event = event
```

By updating the `last_event` variable to be the current position of the mouse, the next time the mouse is dragged, it will draw another line between this point and the next. Your program should look like **paint2.py**. Test it and make sure your paintbrush now works properly.

Change the line width and colour

You only have one colour and thickness for your paintbrush, which limits the drawing you can create. Next, you will amend your GUI so you can pick different colours and line widths.

Add two widgets to the GUI capture a colour and width for the line.

```
from guizero import App, Drawing, Combo,
Slider
```

Add these lines to the app section:

```
color = Combo(app, options=["black",
"white", "red", "green", "blue"])
width = Slider(app, start=1, end=10)
```

paint2.py

> Language: **Python 3**

```

001. # drawing lines by tracking when the mouse is clicked
002.
003. # -----
004. # Imports
005. # -----
006.
007. from guizero import App, Drawing
008.
009. # -----
010. # Functions
011. # -----
012.
013. def start(event):
014.     painting.last_event = event
015.
016. def draw(event):
017.     painting.line(
018.         painting.last_event.x, painting.last_event.y,
019.         event.x, event.y,
020.         color="black",
021.         width=3
022.     )
023.
024.     painting.last_event = event
025.
026. # -----
027. # App
028. # -----
029.
030. app = App("Paint")
031.
032. painting = Drawing(app, width="fill", height="fill")
033.
034. painting.when_left_button_pressed = start
035. painting.when_mouse_dragged = draw
036.
037. app.display()

```

“ You only have one colour and thickness for your paintbrush, which limits the drawing you can create ”

Add ovals

When first creating the Paint application, you used ovals to draw dots across the screen. Can you modify your program to draw ovals again, using a similar process to how rectangles are drawn? Hint: see the **10-paint.py** listing, which also styles up the tools and aligns them neatly in a box.

The Drawing widget also supports drawing triangles and polygons. Check out the documentation (lawsie.github.io/guizero/drawing) and see how you might use this function to create other shapes.

You may also want to change the background colour of your painting to be different. Also, we have used a Combo and a Slider, but you could choose different widgets.

```

        painting.line(
            painting.last_event.x, painting.
last_event.y,
            event.x, event.y,
            color=color.value,
            width=width.value
        )

```

Test your code, resembling **paint3.py** (overleaf), and you can now select the colour and line width.

Drawing shapes

You are going to extend your paint application so you can draw filled rectangles. When the mouse is pressed, the rectangle will appear and grow as the mouse is dragged across the screen. When the mouse button is released, the rectangle will drawn onto the screen.

To do this, you will modify your program to continuously draw and delete rectangles until the mouse button is released. Let's add a widget to your GUI so you can select whether to draw a line or a rectangle. Add this to the app section:

```

shape = Combo(app, options=["line",
"rectangle"])

```

Modify the draw function to only draw lines if the **"line"** option is selected.

```

if shape.value == "line":

```

Figure 2



▲ Figure 2 Not a very good paintbrush

paint3.py

> Language: Python 3

```
001. # widgets to set the color and width
002.
003. # -----
004. # Imports
005. # -----
006.
007. from guizero import App, Drawing, Combo, Slider
008.
009. # -----
010. # Functions
011. # -----
012.
013. def start(event):
014.     painting.last_event = event
015.
016. def draw(event):
017.     painting.line(
018.         painting.last_event.x,
019.         painting.last_event.y,
020.         event.x, event.y,
021.         color=color.value,
022.         width=width.value
023.     )
024.     painting.last_event = event
025.
026. # -----
027. # App
028. # -----
029.
030. app = App("Paint")
031.
032. color = Combo(app, options=["black", "white", "red",
033.                             "green", "blue"])
034. width = Slider(app, start=1, end=10)
035. painting = Drawing(app, width="fill", height="fill")
036.
037. painting.when_left_button_pressed = start
038. painting.when_mouse_dragged = draw
039.
040. app.display()
```

paint4.py

> Language: Python 3

```
001. # adding different drawing shapes
002.
003. # -----
004. # Imports
005. # -----
006.
007. from guizero import App, Drawing, Combo, Slider
008.
009. # -----
010. # Functions
011. # -----
012.
013. def start(event):
014.     painting.last_event = event
015.     painting.first_event = event
016.     painting.last_shape = None
017.
018. def draw(event):
019.     if shape.value == "line":
020.         painting.line(
021.             painting.last_event.x,
022.             painting.last_event.y,
023.             event.x, event.y,
024.             color=color.value,
025.             width=width.value
026.         )
027.     if shape.value == "rectangle":
028.
029.         if painting.last_shape is not None:
030.             painting.delete(painting.last_shape)
031.
032.             rectangle = painting.rectangle(
033.                 painting.first_event.x,
034.                 painting.first_event.y,
035.                 event.x, event.y,
036.                 color=color.value
037.             )
038.             painting.last_shape = rectangle
039.
040.             painting.last_event = event
041.
042. # -----
043. # App
044. # -----
045.
046. app = App("Paint")
047.
048. color = Combo(app, options=["black", "white", "red",
049.                             "green", "blue"])
050. width = Slider(app, start=1, end=10)
051. shape = Combo(app, options=["line", "rectangle"])
052. painting = Drawing(app, width="fill", height="fill")
053.
054. painting.when_left_button_pressed = start
055. painting.when_mouse_dragged = draw
056.
057. app.display()
```

```

        painting.line(
            painting.last_event.x,
            painting.last_event.y,
            event.x, event.y,
            color=color.value,
            width=width.value
        )

```

Test your program to make sure that the line still works and nothing happens when "rectangle" is selected.

Create two new variables to keep track of the first event and the last shape drawn when the mouse button is pressed.

```

def start(event):
    painting.last_event = event
    painting.first_event = event
    painting.last_shape = None

```

These variables will be used when drawing and deleting the rectangle before the mouse button is released.

“ The program will continually draw a rectangle, then delete it ”

```

if shape.value == "rectangle":


    if painting.last_shape is not None:
        painting.delete(painting.last_shape)

    rectangle = painting.rectangle(
        painting.first_event.x,
        painting.first_event.y,
        event.x, event.y,
        color=color.value
    )

    painting.last_shape = rectangle

```

The program will continually draw a rectangle, then delete it, then draw it again until you release the button.

Your complete program should look similar to **paint4.py**. Have fun trying it out – what pictures can you create? 

Custom events

To get your paint application to react to the mouse position, you have used custom events. The events work in a similar way to the normal widget command parameter in that you set them to a function, which is called when that event occurs.

When your function is called, a variable is passed which contains information about the event that has occurred, such as the x and y co-ordinates of the mouse. Most widgets, including the App itself, support the following events:

- when clicked – `when_clicked`
- when the left mouse button is pressed – `when_left_button_pressed`
- when the left mouse button is released – `when_left_button_released`
- when the right mouse button is pressed – `when_right_button_pressed`
- when the right mouse button is released – `when_right_button_released`
- when a key is pressed – `when_key_pressed`
- when a key is released – `when_key_released`
- when the mouse enters a widget – `when_mouse_enters`
- when the mouse leaves a widget – `when_mouse_leaves`
- when the mouse is dragged across a widget – `when_mouse_dragged`

These events can be used to make your GUIs more interactive.

10-painting.py

> Language: Python 3

```

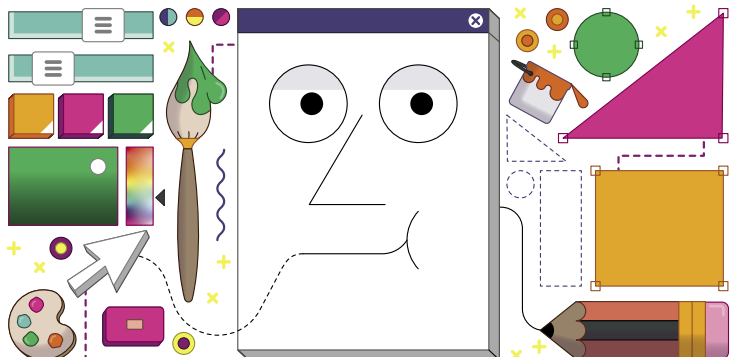
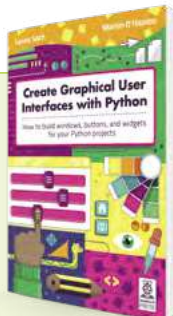
001. # styled up
002.
003. # -----
004. # Imports
005. # -----
006.
007. from guizero import App, Drawing, Combo, Slider, Box, Text
008.
009. # -----
010. # Functions
011. # -----
012.
013. def start(event):
014.     painting.last_event = event
015.     painting.first_event = event
016.     painting.last_shape = None
017.
018. def draw(event):
019.     if shape.value == "line":
020.         painting.line(
021.             painting.last_event.x, painting.last_event.y,
022.             event.x, event.y,
023.             color=color.value,
024.             width=width.value
025.         )
026.
027.     else:
028.         if painting.last_shape is not None:
029.             painting.delete(painting.last_shape)
030.
031.         if shape.value == "rectangle":
032.
033.             painting.last_shape = painting.rectangle(
034.                 painting.first_event.x,
035.                 painting.first_event.y,
036.                 event.x, event.y,
037.                 color=color.value
038.
039.                 if shape.value == "oval":
040.
041.                     painting.last_shape = painting.oval(
042.                         painting.first_event.x,
043.                         painting.first_event.y,
044.                         event.x, event.y,
045.                         color=color.value
046.                     )
047.
048.                     painting.last_event = event
049. # -----
050. # App
051. # -----
052.
053. app = App("Paint")
054. app.font = "impact"
055.
056. tools = Box(app, align="top", width="fill", border=True)
057.
058. Text(tools, text="Tool", align="left")
059. shape = Combo(tools, options=["line", "rectangle",
060.                               "oval"], align="left")
061.
062. Text(tools, text="Colour", align="left")
063. color = Combo(tools, options=["black", "white", "red",
064.                               "green", "blue"], align="left")
065.
066. Text(tools, text="Width", align="left")
067. width = Slider(tools, start=1, end=10, align="left")
068.
069. painting = Drawing(app, width="fill", height="fill")
070.
071. painting.when_left_button_pressed = start
072. painting.when_mouse_dragged = draw
073.
074. app.display()

```

Create Graphical User Interfaces with Python

For further tutorials on how to make your own GUIs with guizero, take a look at our book, *Create Graphical User Interfaces with Python*. Its 156 pages are packed with essential info and a range of exciting projects.

magpi.cc/pythongui



9,8 MILLION+ PRODUCTS ONLINE | 1,900+ INDUSTRY-LEADING SUPPLIERS | 100% FRANCHISED DISTRIBUTOR

Enabling the World's Ideas **DIGIKEY.CO.UK**

Bluetooth

5G

IoT

Automation

Wireless

**FREE
SHIPPING**
ON ORDERS OVER
£33 OR \$50 USD*



*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of \$18.00 USD will be billed on all orders of less than \$50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2021 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

Raspberry Pi Pico: digital communication protocols

Explore the I2C and SPI communication protocols and use them to display data on an LCD



Ben Everard

Ben Everard is a geek who has stumbled into a career that lets him play with new hardware. He's also the editor of HackSpace magazine (hsmag.cc).

[@beneverard](https://twitter.com/beneverard)

In previous Raspberry Pi Pico tutorials we've looked at how to work with a few common bits of hardware, but as you build more projects on your own, you'll probably want to branch out to use all sorts of different sensors, actuators, and displays. How will you communicate with these? Sometimes you may find that there's a MicroPython library you can use where someone's already converted the low-level functions into an easy-to-use interface. However, this isn't always the case.

Fortunately, there are a couple of standard ways of connecting low-level digital devices together that are implemented in MicroPython: Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI). In many ways, they're very similar in that they both define a way of wiring up a two-way interface between two devices. In fact, many parts come in versions with either interface, so you can pick the one that's right for your project. In both cases, there's one device that controls the communication (your Pico) and one (or more) that waits for instructions from the main device. However, there are a few differences. We'll look at both protocols now and then we'll help you pick the right one for each project.

I2C protocol

Communication over I2C takes place on two wires: a clock (usually marked as SCL) and a data channel (usually marked SDA).

These have to connect to specific pins on the Pico. There are a few choices; take a look at the pinout diagram for the options (**Figure 1**). There are two I2C buses (I2C0 and I2C1), and you can use either or both. In our example, we'll use I2C0 – with GPO for SDA, and GP1 for SCL.

To demonstrate the protocols, we'll use a SerLCD module from SparkFun. This has the

advantage that it has both I2C and SPI interfaces, so we can see the differences between the two methods with the same hardware.

This LCD can display two lines, each with up to 16 characters. It's a useful device for outputting bits of information about our system. Let's take a look at how to use it.

Wiring I2C is just a case of connecting the SDA pin on the Pico with the SDA pin on the LCD and the same for the SCL. Because of the way I2C handles communication, there also needs to be a resistor connecting SDA to 3.3 V and SCL to 3.3 V. Typically these are about 4.7 kΩ. However, with our device, these resistors are already included, so we don't need to add any extra ones.

With this wired up (**Figure 2**, overleaf), showing information on the screen is as simple as:

```
import machine
sda=machine.Pin(0)
scl=machine.Pin(1)
i2c=machine.I2C(0,sda=sda, scl=scl,
freq=400000)
i2c.writeto(114, '\x7C')
i2c.writeto(114, '\x2D')
i2c.writeto(114, "hello world")
```

This code doesn't do very much. It connects to the I2C device and sends some data. However, there are a few bits that may look a little unusual.

The 114 in the `i2c.writeto()` lines refers to the address of the I2C device. You can connect many devices to an I2C bus (more on this later), and each time you want to send or receive data, you need to specify the address of the device you want to communicate with. This address is hard-wired into the device (though you may be able to change it by cutting a trace on the PCB, or soldering a blob – see your device's documentation for details).

You'll Need

- Raspberry Pi Pico
- SerLCD module
- Jumper wires

You should find the address for your device in the documentation, but you can scan an I2C bus to see what addresses are currently in use. After setting up the I2C bus, you can run the `scan` method to output the addresses currently in use:

```
import machine
sda=machine.Pin(0)
scl=machine.Pin(1)
i2c=machine.I2C(0,sda=sda, scl=scl,
freq=400000)
print(i2c.scan())
```

“ This LCD can display two lines, each with up to 16 characters ”

The next bits that may look a little odd are the `\x7C` and `\x2D` commands that are written. Each I2C device requires data sent in a specific format. There's no standard for this, so you'll have to refer to the documentation for whatever I2C device you're setting up. The `\x` at the start of each of these tells MicroPython that we're sending a hexadecimal string, which is a common way of ensuring you're sending the exact data you want. For our LCD, `7C` enters command mode and `2D` blanks the LCD and sets the cursor to the beginning. Following this, we can send data that's displayed on the screen:

```
file.close()
```

Of course, there's not much use in a screen that just says Hello World, so let's take a look at turning this into something a little more useful – a thermometer. In issue 108 (magpi.cc/108), you learned how to use the ADC to read temperatures using your Pico's internal temperature sensor. We can now build on this code to make a standalone thermometer that doesn't need a computer to read the output. With your LCD still connected as before, run the following code:

```
import machine
import utime

sda=machine.Pin(0)
scl=machine.Pin(1)
i2c=machine.I2C(0,sda=sda, scl=scl,
freq=400000)
```

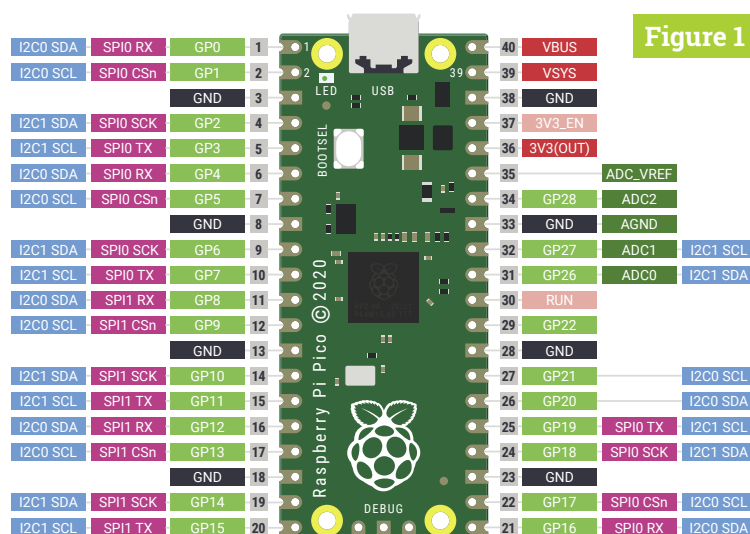


Figure 1 The pinout for Raspberry Pi Pico. I2C functions are shown in light blue; SPI in pink

```
adc = machine.ADC(4)
conversion_factor = 3.3 / (65535)
while True:
    reading = adc.read_u16() * conversion_
    factor
    temperature = 25 - (reading -
0.706)/0.001721
    i2c.writeto(114, '\x7C')
    i2c.writeto(114, '\x2D')
    out_string = "Temp: " + str(temperature)
    i2c.writeto(114, out_string)
    utime.sleep(2)
```

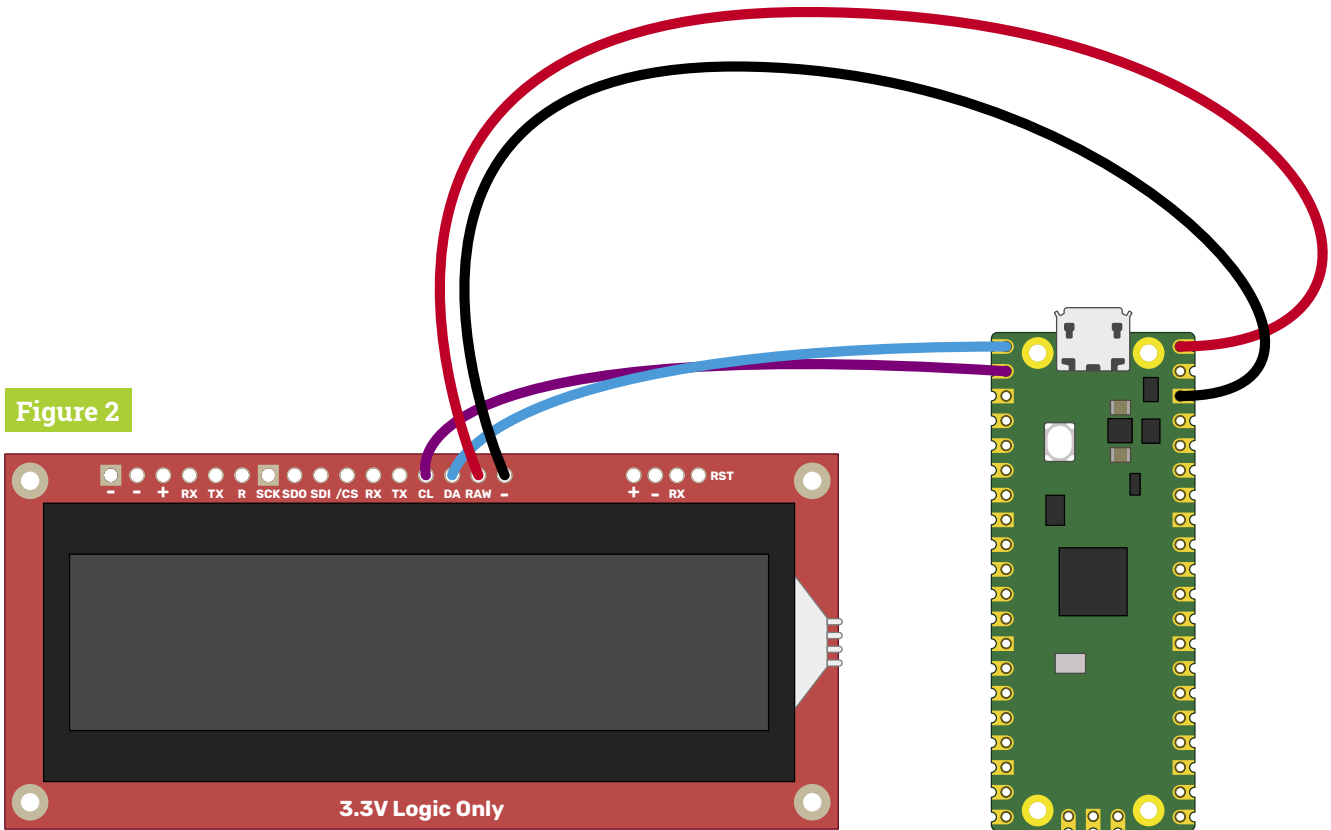
This should mostly look familiar. The only slight change to the previous temperature code is that before we outputted the result of our calculation – a number – but the LCD needs characters to display, so we use the `str` function which converts the number to a string of characters. We can then build this into a slightly more informative output by combining it with `"Temp: "`.

As you've seen, I2C is an easy way of linking extra hardware to your Pico. You will need to ensure you've got appropriate documentation for whatever device you want to connect that lets you know what commands do what, but as long as you know this, you can easily add all sorts of bits and bobs to your Pico and create impressive builds.

Serial Peripheral Interface

We've seen how I2C works, now let's take a look at SPI. We'll use the exact same LCD, so the commands and everything else are the same, it's just the protocol we send data over that's different. SPI has four connections: SCLK, MOSI, MISO, and CS (sometimes labelled SS). SCLK is the clock, MOSI is the line taking data from your Pico to the peripheral device, and MISO takes data from your

Figure 2



▲ Figure 2 Wiring up a SerLCD module for I2C

peripheral device to your Pico. CS stands for Chip Select and is used to connect many devices to a single SPI bus. You simply have to apply power to the CS line to enable an SPI peripheral and pull it low to disable it. To confuse things slightly, this particular device doesn't have CS, but /CS which stands for NOT CS – in other words, it's the opposite of CS, so you bring it low to enable the LCD and high to disable it. You could connect the CS to a GPIO pin and toggle this on and off to enable and disable the display, but since we only have one device, we can simply connect it to ground to keep it enabled (Figure 3).

So, with the SerLCD's power lines connected to VBUS and GND, we just need to connect its SDO to Pico's MISO (GP4 / SPIo RX), SDI to MOSI (GP3 / SPIo TX), SCK to SCLK (GP2 / SPIo SCK), and /CS to GND. There are no addresses in SPI, so we can just dive in and write our code:

```
import machine

spi_sck=machine.Pin(2)
spi_tx=machine.Pin(3)
spi_rx=machine.Pin(4)

spi=machine.SPI(0,baudrate=100000,sck=spi_sck,
mosi=spi_tx, miso=spi_rx)
spi.write('\x7C')
spi.write('\x2D')
```

```
spi.write("hello world")
```

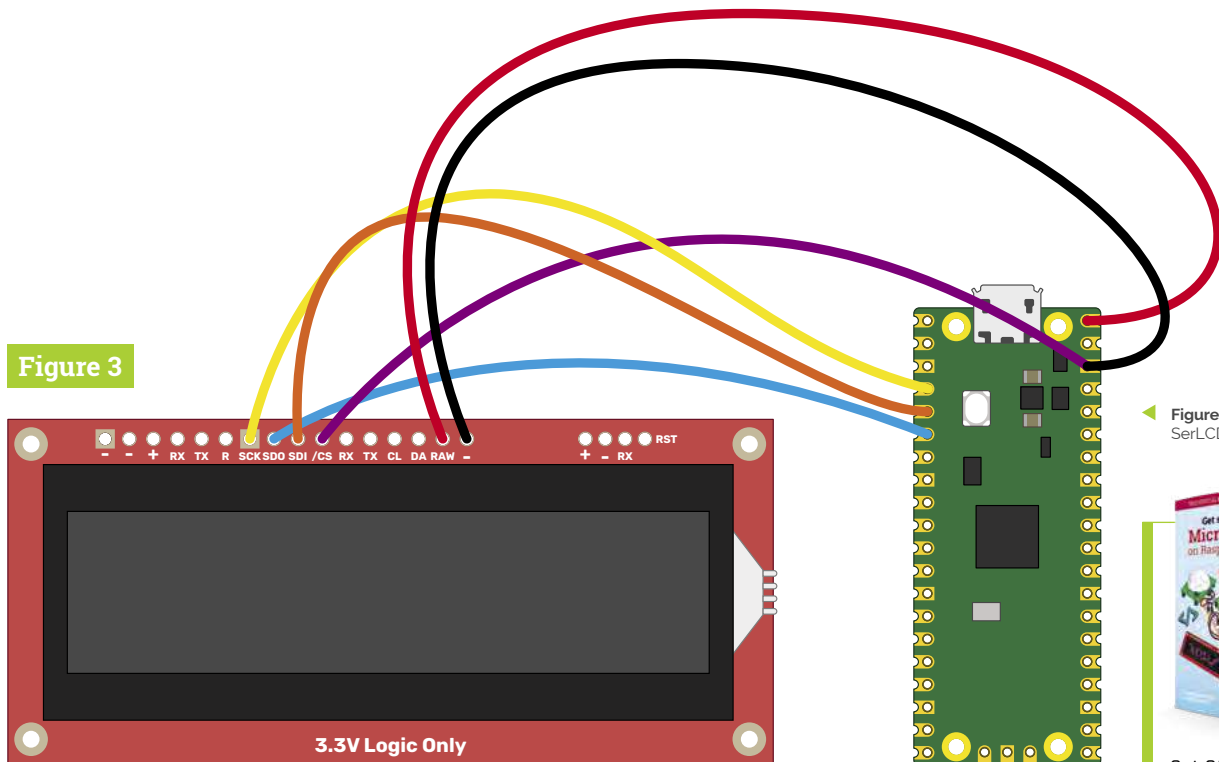
In this case, we're using SPIo, and one set of available pins for this is GP2, GP3, and GP4. Most types of serial communication have a speed or baudrate, which is basically how many bits of data it can push through the channel per second. A lot of things affect this, such as the capabilities of the two devices being connected and the wiring between them (how long it is and if there's interference from other devices). If you find you're having problems with mangled data, then you may need to reduce it. For our little screen, we're just sending one byte of data per character, so it doesn't really matter how fast we send it, but for some other SPI devices (such as pixel-based displays), fine-tuning the baud rate can be important.

Let's have take a look at how this leaves our thermometer code:

```
import machine
import utime

spi_sck=machine.Pin(2)
spi_tx=machine.Pin(3)
spi_rx=machine.Pin(4)

spi=machine.SPI(0,baudrate=100000,sck=spi_sck,
mosi=spi_tx, miso=spi_rx)
```

**Figure 3****Figure 3** Wiring up a SerLCD module for SPI

Get Started with
MicroPython
on Raspberry
Pi Pico

For more physical computing projects to try on your Raspberry Pi Pico, grab a copy of the new book, *Get Started with MicroPython on Raspberry Pi Pico*. As well as learning how to use Raspberry Pi Pico's pins as inputs and outputs, you'll build a simple game, measure temperatures, save and load data to your Pico's file system, and even make a burglar alarm for your room. *Get Started with MicroPython on Raspberry Pi Pico* is available now from magpi.cc/picobook.

```
adc = machine.ADC(4)
conversion_factor = 3.3 / (65535)

while True:
    reading = adc.read_u16() * conversion_factor
    temperature = 25 - (reading -
0.706)/0.001721
    spi.write('\x7C')
    spi.write('\x2D')
    out_string = "Temp: " + str(temperature)
    spi.write(out_string)
    utime.sleep(2)
```

As you can see, there's really very little difference in the code between I2C and SPI. Once you've got everything set up, the only really change is that with I2C you have to specify the address when you send data, while with SPI you don't (though remember if you had more than one device attached, you'd need to toggle the CS GPIO to select the appropriate device).

So, if they're so similar, which protocol should you choose when building a project? There are a few

“ For many projects,
you can happily use
either protocol ”

factors to consider. The first is availability of the things you want to attach. Sometimes a sensor is only available as I2C or SPI, so you have to use that. However, if you've got a choice of hardware, the biggest impact comes when you're using multiple extra devices. With I2C, you can connect up to 128 devices to a single I2C bus; however, they all need to have a separate address. These addresses are hard-wired in. Sometimes it's possible to change the address with a solderable (or cuttable) connection, but sometimes it's not. If you want to have multiple of the same type of sensors (for example, if you're monitoring the temperature at many points on your project), you may be limited by the number of I2C addresses for your sensor. In this case, SPI may be a better choice.

Alternatively, SPI can have an unlimited number of devices connected; however, each one has to have its own CS line. On the Pico, there are 26 GPIO pins. You need three of them for the SPI bus, so that means there are 23 available for CS lines. And this is assuming you don't need any for anything else. If available GPIOs are at a premium, then you may want to look at I2C.

In reality, for many projects, you can quite happily use either protocol, and you may find that the choice of which to use has more to do with what parts you find in your parts box than a technical difference between the two. ■

Isomorphic keyboard: hex keys



Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

Use the Pico board to make an isomorphic music keyboard with hexagons

Last month we showed you how to make a switch matrix and enclosure for an isomorphic layout keyboard. This month we will finish the construction, and look at the Pico board, control switches, OLED display, and rotary encoder, as well as how to wire it all together. We will also look at the software we need to test what we have built, and the final software to bring it all to life.

01 The control keys

We added four triangular control keys to change various non-playing parameters, and built them on a separate piece of stripboard from the main matrix. They are, however, wired into the main matrix as an extra column, number 14. **Figure 1** shows the physical layout diagram of this. Note that the top left-hand corner of the board has been marked with a small white mark, as we found it helped us orientate the board while adding the components. During play, these are used to switch the MIDI channel and shift the octave of the keyboard.

02 The OLED display

We used a 128 by 64 OLED I2C display to provide visual feedback on what notes were being sent. You can use any colour, but we used the type

with a strip at the top in one colour, and the rest in another. The specific one we used had yellow LEDs for the top 16 rows, with the rest of the display blue. Like our other Pico projects, we fitted a right-angled header on the display, pointing inwards to give a low profile mounting, and milled out half the thickness of the box to accommodate the display's bezel and connector strip.

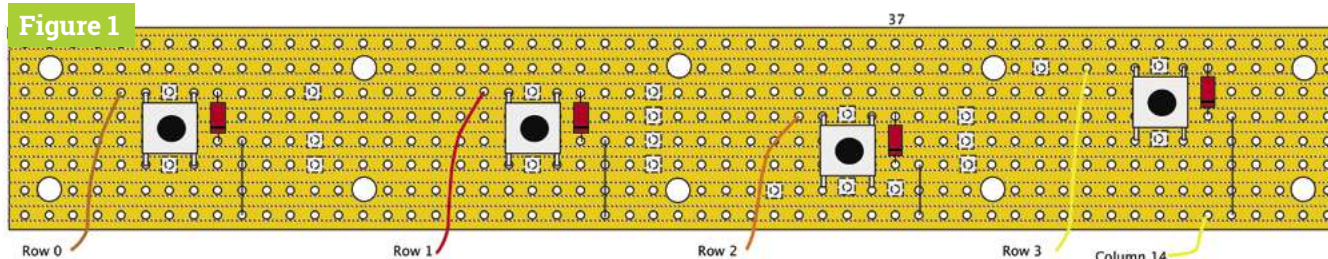
03 Rotary encoder

We added our normal KY-040 rotary encoder to interact with the display and control the Note On velocity during play. Due to the library we used to read it, the clock and data signals needed swapping round in order to get the rotation correct – that is clockwise rotation producing an increasing count. **Figure 2** shows how we did this by swapping over the two connectors in the ribbon cable. Note the green and yellow wires cross so we can keep the interface connectors the same as we normally use on our other projects.

04 The Pico board

The whole keyboard is driven by the Pico controller board, and the schematic is shown in **Figure 3** (overleaf). The OLED and rotary encoder are directly connected to the Pico, as are the row inputs of the matrix. These are connected to a 4-,

Figure 1



DOWNLOAD
THE FULL CODE:

 magpi.cc/pibakery



Display showing current note, octave shift, channel, and note velocity

Rotary knob for note velocity and menu navigation

5-, and 8-pin header respectively. The matrix column signals, on the other hand, are generated by two 74HS138 decoder chips, because otherwise we would not have enough Pico pins. Each chip produces a logic zero on the output determined by three binary bits input. By using an extra two pins, we can enable one decoder or the other.

05 The Pico board construction

Figure 4 shows the physical layout of the Pico board. Note that here we are not only using the normal Pico pins, but test pad 6 to bring out the bootloader switch to the back panel. We knew in advance that there would be a lot of dismantling to be done if we ever needed to press it, and we were very glad when we found it necessary during development. **Figure 5** shows the point on the

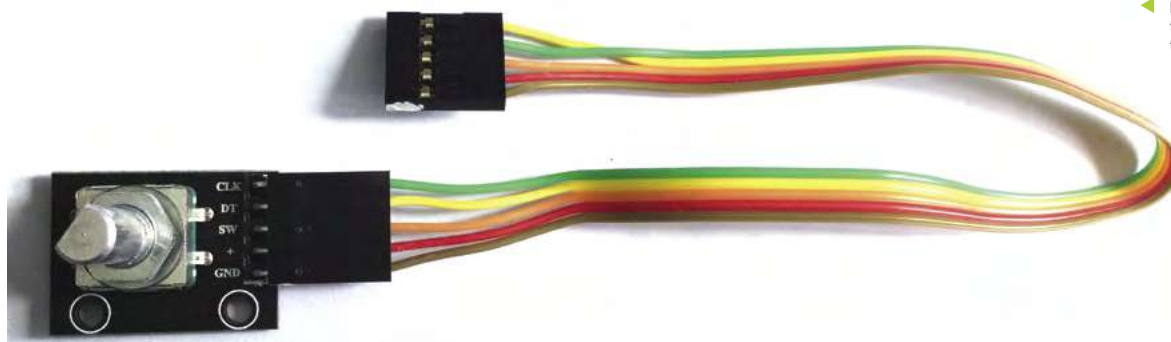
back of the Pico board to solder this wire. Note the column output connection order was chosen to simplify the wiring of this board.

06 Wiring the matrix

We used ribbon cable with crimped headers to do all the interconnections. As we only had 21 cm lengths, we needed to use two lengths to make them long enough to fit for assembly. They were joined together with a strip of long pin headers and, as shown in **Figure 6**, we needed to use two back shells to make up the required number of column pins. These connectors have to live inside the lower processor enclosure after assembly, so some of the wires had to be extended by a few centimetres in order to reach where they were going on the matrix board.

You'll Need

- ▶ OLED 128×64 I2C display SSD 1306 yellow/blue magpi.cc/piboled
- ▶ KY-040 rotary switch magpi.cc/pibky040
- ▶ Momentary push switch magpi.cc/pibmomentary



◀ **Figure 2** Swapping over the clock and data in the encoder cable

Figure 2

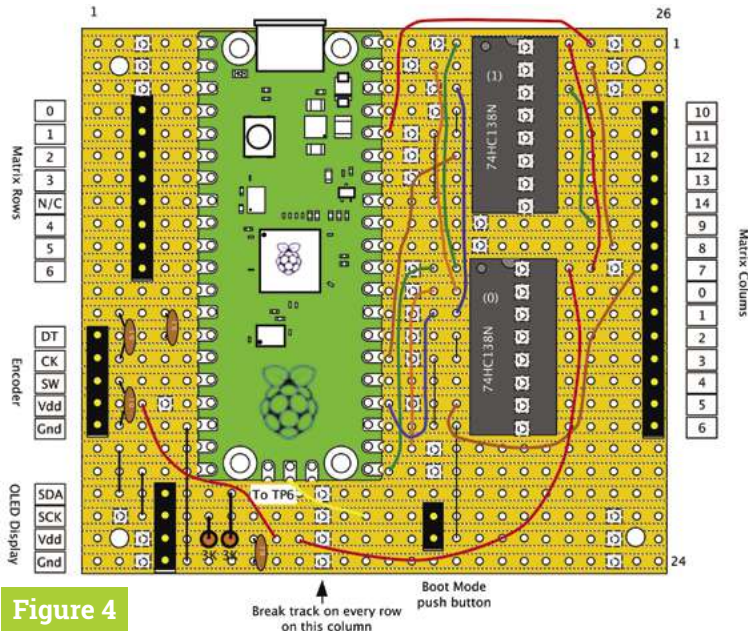


Figure 4

▲ **Figure 4** Physical layout of the Pico controller

Top Tip

Fitting the lid

If you find a key will not click when pushed down, you probably have a wire that has moved over a key. Remove the top and rearrange the wires.

▼ **Figure 3** Schematic of the Pico controller

07 Test all the parts

Before engaging in the final assembly, make sure all the sections work. **Figure 7** shows the bottom section with the Pico processor and leads. Connect the display and encoder to the ribbon cables, but don't mount them on the top yet. Also connect up the column and row connectors to the matrix. Then hop over to our GitHub page and run the examples to test the encoder, display, and matrix. With the matrix, check that every key gives a key number. If any key shows repeating key numbers, it is likely that breaks have not been fully made in the wiring around those keys.

08 Mounting the top

With the OLED display fixed into place and the rotary encoder bolted on, you can now attach

the top. Put the top side ring into place, and make sure all the ribbon cables for the matrix are tucked inside the lower box (**Figure 8**). We found it best to fit three columns at once, turn them over while pressing a board over the top, and then tape them up. We did try turning the whole lot of keys over twice, but a lot fell out. Do this with the top on a thick book so you can get it flat with the encoder shaft not fouling it.

09 Final assembly

Now attach the ribbon cables to the display and the encoder, and place the top on the side rings. Just put one of the countersunk screws in the top and just engage the thread, but don't screw it up yet. Make sure no wires are sticking out of the sides; push them back if they are. Make sure the side ring is clicked into the middle tabs, and then tighten up the top screw. Insert the other screws and tighten them up carefully, checking no wires poke out and the tabs remain engaged. Test that all the buttons click when pressed.

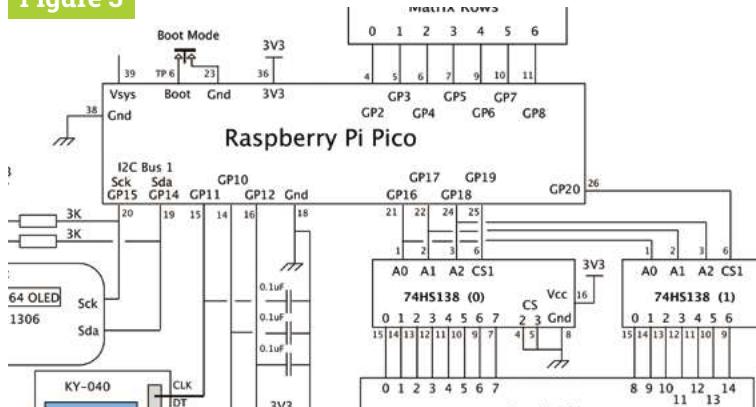
“ Before engaging in the final assembly, make sure all the sections work ”

Figure 5



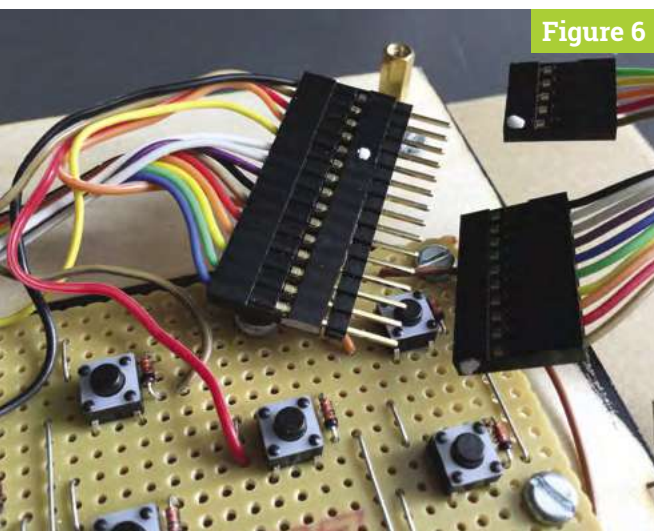
▲ **Figure 5** Accessing the bootloader signal

Figure 3



10 The software

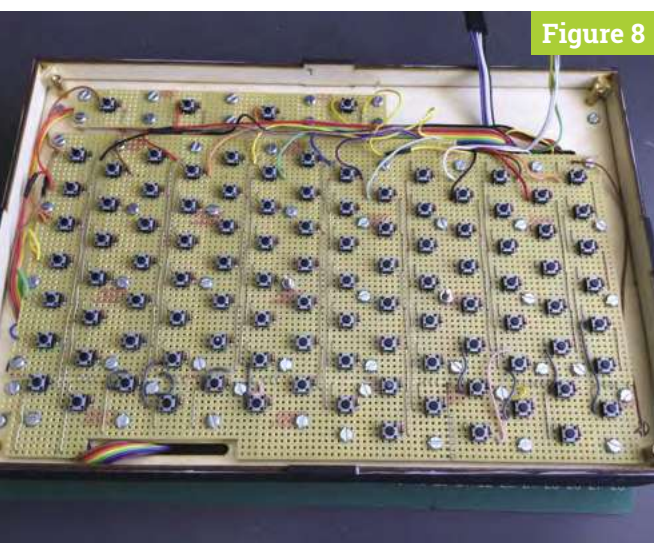
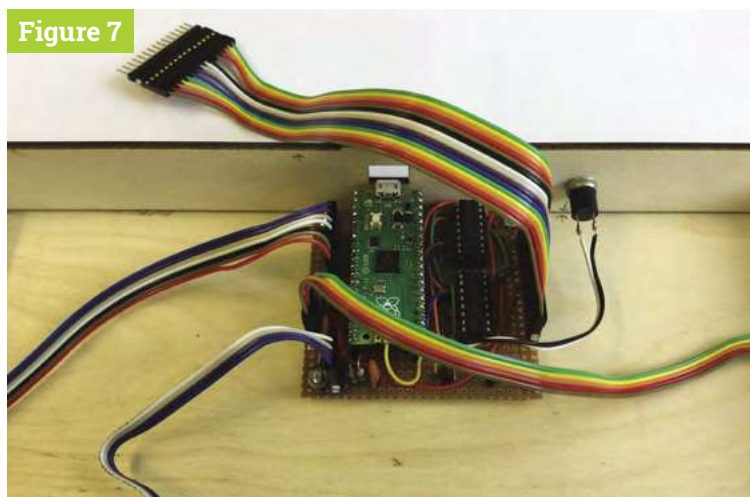
We decided to use CircuitPython from Adafruit, having used MicroPython, and C in our previous two projects. It had the big advantage of having a USB MIDI capability, but does not have any interrupt handling capability. This was nearly a showstopper until we found an example of using a rotary encoder using the GPIO pins. We found the OLED driver complex to understand, especially for the simple display we wanted to

**Figure 6**

produce, and the examples we found weren't all that comprehensive. But, in the end, we prevailed. See the implementation notes on our GitHub page for details of how to set up your Pico.

11 Play mode

In the play mode, the note name and octave number are displayed when you hold down a key; when more than one is held down, the last key is shown. The buffer can cope with up to ten keys being held down at once. The left/right triangle controls change the channel, and the up/down ones change the octave. The rotary encoder selects the note-on velocity. When playing Channel 10, the percussion names are displayed. To get into the instrument select mode, you push the switch incorporated into the rotary encoder.

**Figure 8****Figure 7**

12 Instrument Select mode

Once in the Select mode, a list of instrument groups is shown. Turn the encoder to highlight the required group, then push the encoder knob again, and you are taken to the first instrument in that group, in a list of all 128 instruments that you can scroll through. Push the encoder to send a MIDI message to change the instrument, and the left/right controls to change the channel. The up control takes you first to the instruments and then back into play mode. Note that, once in the Select mode, the down button acts the same as pushing the encoder button.

◀ **Figure 6** Connecting the matrix column signals to the Pico on the bottom layer

▲ **Figure 7** The Pico board ready for connection to the peripheral devices

Top Tip

Other chords

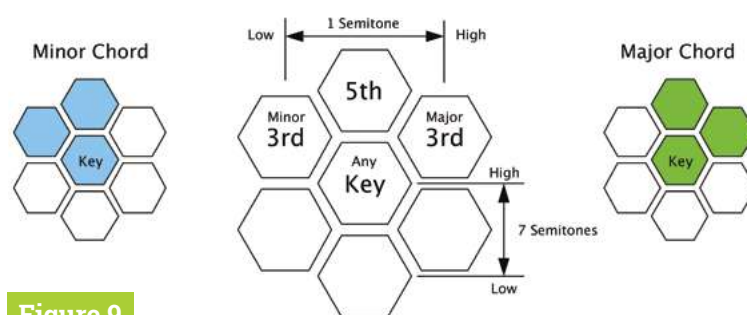
For more chord shapes, see magpi.cc/chordshapes.

◀ **Figure 8** The top layer waiting for the keys to be placed

▼ **Figure 9** Note intervals and simple chords with the Hex-Keys layout

13 Chords you can play

Each note appears twice on the keyboard, left and right side. Given any starting note, the note above it is a perfect fifth up, to the left is a minor third, and to the right is a major third. So you can play a major or minor triad, as shown in **Figure 9**. Chords shapes are the same for any chords in any key. This is reputed to be much easier to learn than a conventional piano keyboard, and once you know how to play one chord, those in other keys are the same shape, but just transposed. [M](#)

**Figure 9**



THE *Official* RASPBERRY PI HANDBOOK 2021



Radio
Globe



Magic
Mirror



Reachy
Robot



Hot
Spotter Drone



High Quality
Camera



LED
Mirror



Dual-screen
Instaclock

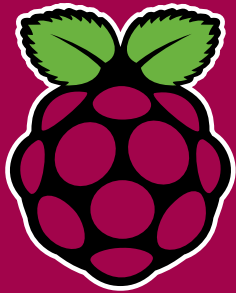


Introducing
RASPBERRY PI 400

Make amazing things
with Raspberry Pi

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

THE *Official*



RASPBERRY PI HANDBOOK

2021



200 PAGES OF RASPBERRY PI

- Get started with Raspberry Pi, electronics, and more
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store

RASPBERRY PI Student Guide

A computer with web access is an essential piece of back to school kit. **Rosie Hattersley** looks at how students can equip themselves for the year ahead

Starting the new school year without a computer is unthinkable these days.

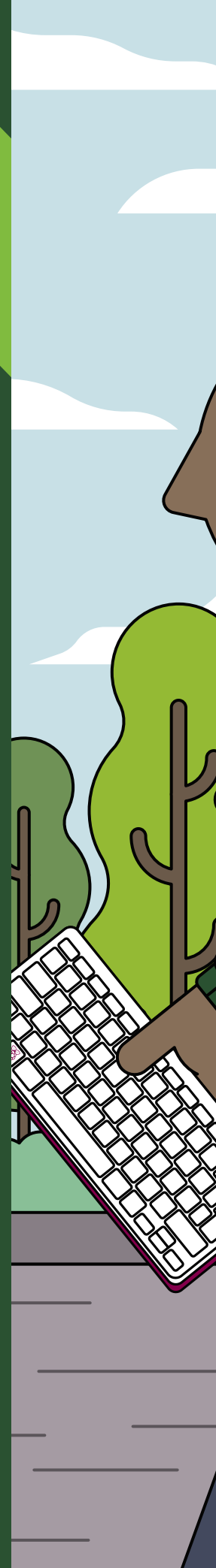
We've come to rely on them for everything, from recording notes and writing up the results of our school lab experiments, to online research and completing and submitting our homework, not to mention emailing teachers and peers to check exactly what's required in the first place. It's also hard to ignore the trend towards video-based tuition and blended learning, both of which work best with a decent-sized screen on a computer of your very own, rather than a teeny smartphone display or a laptop shared with other family members.

Forking out for multiple laptops gets expensive, which is why Raspberry Pi is such a compelling choice of computer for home study. A 2GB Raspberry Pi 4 costs just £35/\$35, and packs a 64-bit 1.5GHz Broadcom processor

with all the wireless, Bluetooth, and Ethernet connections you need to get online. It even comes complete with its own easy-to-use operating system. Supporting H.264 HD video, it's all geared up for video-based lessons and entertainment duties, and is powerful enough for almost any school or college task. If music or video production or design work are on the curriculum, plump for the even more capable 4GB or 8GB Raspberry Pi 4 model.

To keep costs to a minimum, you could repurpose an unwanted TV with HDMI input and simply add a USB keyboard and mouse – an impressive setup for little more than £50/\$50.

Or, you can get the computer and keyboard in one handy Raspberry Pi 400 package from just £65/\$70. Whichever Raspberry Pi you choose, you can be sure it'll be the ideal computer for school and homework.







DOING HOMEWORK WITH RASPBERRY PI

Independent learning's a breeze with the help of these handy resources

Once you've set up Raspberry Pi, you'll want to investigate apps that make light work of learning. If you're a Microsoft Office aficionado, head straight for LibreOffice, the excellent free office suite that's compatible with Microsoft's paid-for offering. It works in much the same way, meaning a minimal adjustment period. Spreadsheets, databases, a page layout package, and a word processor are all included.

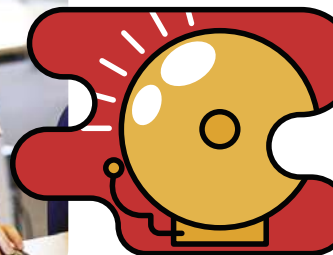
If all things Google appeal, Google Docs and Google Drive – with their automatic syncing across smartphones, laptops, and tablets – are an ideal option for tapping out essays and saving research artefacts. Freshly updated speech recognition tools also make Google's apps a great choice if you like to verbalise your thoughts or use voice commands to initiate searches or dictate documents.

Zoom and Microsoft Teams let you record lectures and lessons (with prior permission from other attendees), which is handy if you need to recap on a point you didn't quite understand. Transcription-wise, you could try Google or YouTube (upload a video to have it transcribed), but if you regularly need talks transcribing, try Otter (otter.ai), which accepts recordings from phones as well as recording live then outputting the conversation as a text file.



▲ Isaac has just been expanded to offer help for GCSE as well as A-level Computer Science courses

GCSE and A-level Computing students can get a helping hand with Isaac, a free online portal designed to aid both students and teachers, with sample lessons and revision modules for your chosen exam board (isaacomputerscience.org).



Supporting schools

Raspberry Pi can supply school kids with computers for use at home so they can join their classmates for online lessons, as well as researching topics in order to complete homework tasks. The Raspberry Pi Foundation provides families with home computers, while partner charity the Bloomfield Trust provides free internet access. The Raspberry Pi Computers For Schools Assistance Programme (magpi.cc/distributingpi) is actively encouraging schools to get in touch with requests for computing kit on behalf of pupils who need one. Should it be needed, the amazing School Home Support team are a call away.



▲ Pupils who need them have been offered a complete Raspberry Pi computing setup to use at home

School Setup

The kit you'll need to use Raspberry Pi for schoolwork

RASPBERRY PI 400

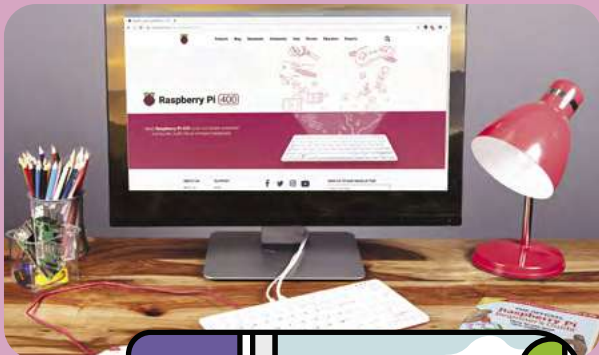
A relative newcomer to the Raspberry Pi line-up, this is a complete 64-bit 4GB RAM computer hidden inside a smart, low-profile keyboard. Just add a mouse and power supply and plug it into any HDMI-equipped screen and you've immediately got a space-saving computer that can be used for homework, online learning, and watching 4K video. A £90/\$100 kit also comes complete with a mouse, official power supply, and 252-page *Beginner's Guide* book.

magpi.cc/raspberrypi400



MONITOR

You can use any HDMI computer or TV screen with Raspberry Pi. Both Raspberry Pi 4 and 400 have two HDMI ports and support dual screens, so you can do your coding on one display and type up your working on the other.



RASPBERRY PI OS

This is the free operating system that works best with Raspberry Pi and runs off a microSD card you plug into Raspberry Pi. Pop it into another computer and download the latest version of the operating system. Apps are sorted into categories such as Internet, Games, and Accessories, and include the powerful LibreOffice productivity suite.

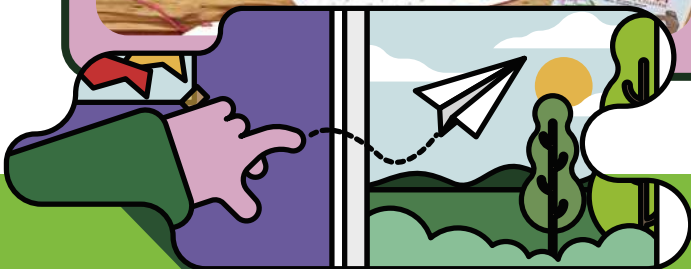
magpi.cc/raspberrypi400

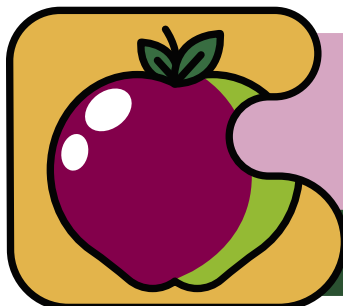


RASPBERRY PI 4

The obvious choice of Raspberry Pi for someone who has an interest in programming, computing, electronics, or robotics, Raspberry Pi 4 (from £35/\$35) boasts incredible processing power for the money and is endlessly customisable thanks to the dozens of add-on HATs that turn it into anything from a music sequencer to an earthquake monitor to the heart of a robot and back to a powerful desktop computer.

magpi.cc/raspberrypi4





LEARN TO CODE WITH RASPBERRY PI

Make like a boss and unlock the magic instructions that tell all those smart devices what to do

Code tells your Raspberry Pi (or anything with a processor inside, such as a smart fridge, printer, or car) what to do.

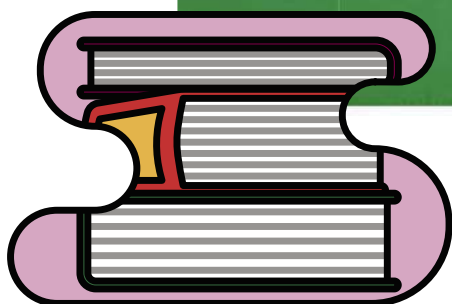
Mastering code means you can go from asking Alexa preset questions to telling an electrical sensor what type of alert to give you when a specific event occurs, and when to leave you in peace. Coding – whether with the entry-level Scratch or the more advanced Python or C++ – is what enables you to tell Raspberry Pi to perform an action such as checking the weather forecast or getting its HQ Camera to take a series of photos of the night sky.

Wannabe coders should head for the Raspberry Pi website, where there's a host of practical projects to complete: projects.raspberrypi.org. Scroll down to see the very latest programming projects to try,

including a Pac-Man-style treasure-hunt guide to using terminal, a website design guide, a role-playing text adventure course, and the chance to work on bigger projects with other people – ideal for classes or after-school clubs.

Older students will benefit both from the Isaac portal mentioned previously, and from using Mathematica (wolfram.com/mathematica) alongside Python, C++, SQL, and so on. It's a great aid for technical subjects from science, engineering and maths to computing, boasting invaluable datasets from around the world that can be used in almost any computer program. For specific computing languages, check out our guides to learning C, Python, and GUI (graphical user interface) programming. Download our free books: magpi.cc/books.

▼ Choose a project you like and learn to code as you go



“ Wannabe coders should head for the Raspberry Pi website, where there's a host of practical projects ”

Three robots that can help bring code to life

Use your Raspberry Pi to control another being

MARTY THE ROBOT V2

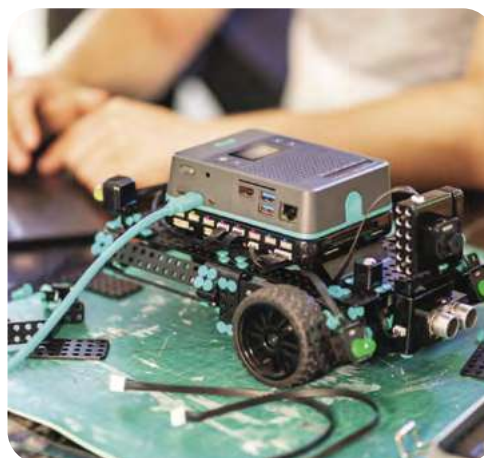
Marty V2 (£314) is a personality-packed coding companion that helps its owner learn how to program him to walk, talk, and interact with people, even shaking hands. An online portal provides tutorials on adding motion detectors to help Marty sense the world around him.



magpi.cc/marty

PI-TOP [4] ROBOTICS KIT

This modular robot kit (£188) works with a pi-top [4] to become an artificially intelligent motorised vehicle to which – thanks to the expansion plate – you can add a wide-angle camera as an all-seeing eye, servos, and motors. Easy to set up, it has a versatile rivet system for building different robots. Powerful with easy-to-use classroom integration.



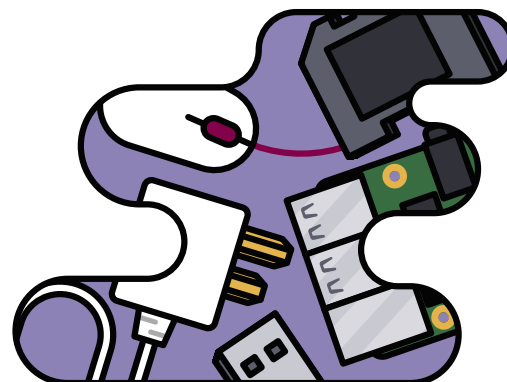
magpi.cc/pitoprobotics

ROCKYBORG

This speedy three-wheeler (£100) corners like a dream and is an ideal vehicle [sorry!] for learning how to build and control a robot that has servos and a geared motor. Add-ons include a Raspberry Pi camera to capture the driver's eye view.



magpi.cc/rockyborg



Sign up for a Raspberry Pi challenge

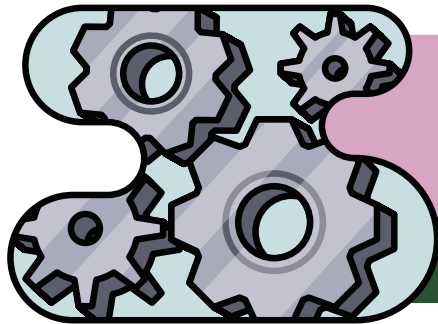
There are lots of great Raspberry Pi events that take place throughout the year that can help you get even more from this amazing little computer. If you're a big fan of all things space-related, you should definitely get on board the Astro Pi project (astro-pi.org). Run in conjunction with the European Space Agency, it gives students a direct means of sending a message to astronauts aboard the ISS – using code, of course – as well as challenging you and your school to submit an experiment that can be run by them up in space!

If action is more your thing, sign up for Pi Wars (piwars.org) and get designing and building your own robot. It's a Raspberry Pi version of TV's *Robot Wars*, with multiple obstacle courses and an incredibly competitive Pi Noon shoot-out that draws competitors from around the world.

If Raspberry Pi itself – or making things in general – inspires you, see whether there's a Code Club (codeclub.org) or CoderDojo (coderdojo.com) in your area. If you're really keen, why not volunteer to help run one and share your enthusiasm and skills with your peers?



▲ Could you design your own robot to take on the Pi Wars obstacle course?



LEVEL UP YOUR SKILLS WITH RASPBERRY PI

Get creative with these rewarding projects. Students can learn incredible skills (and code) while building useful projects

PLAY MUSIC WITH RASPBERRY PI

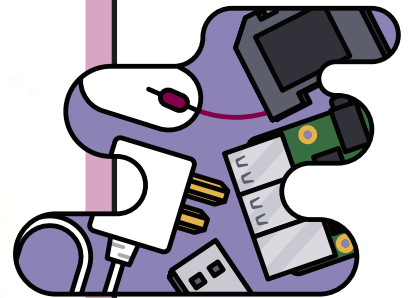
Raspberry Pi can be used as a musical instrument or synthesizer that deepens students' understanding of music theory. You could start with one of the musical challenges on Raspberry Pi's projects page (magpi.cc/musicprojects), and become a guitar hero or an amazing drummer – all using code, of course.

If you're keen to ensure your sweet sounds directly reference the official curriculum, pop a Sense HAT on your Raspberry Pi and try the Scratch MP3 player challenge (magpi.cc/scratchmp3). You'll use basic programming constructs and digital, analogue, and electromechanical components to get visual as well as aural feedback on your code-based mixes.

For an interactive mixing tool, try the free Sonic Pi (magpi.cc/sonicpi) in which coding is used to create sounds, often using everyday objects for the samples.



▲ Experiment with Sonic Pi's code-based sound mixing features

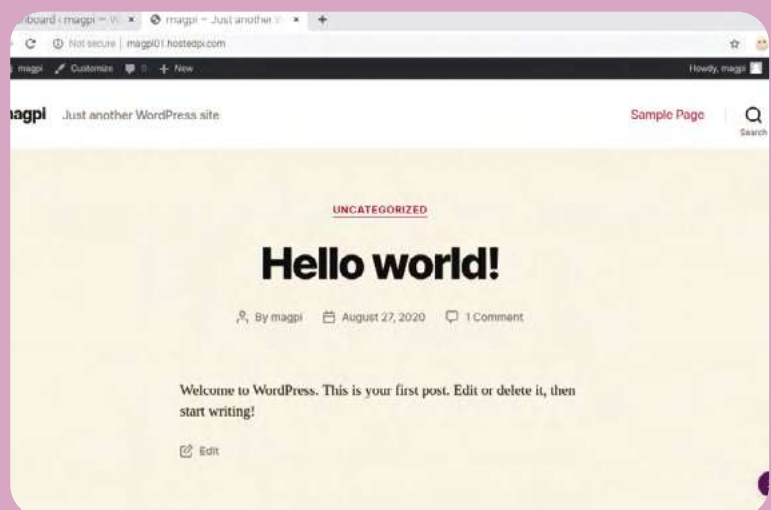


WEB DESIGN

Creating a blog or a real website can easily be achieved using your Raspberry Pi computer. Both projects are included in the National Curriculum. You can either type the content straight into web pages, or use the text editor built into Raspberry Pi OS and then copy and paste it all into the relevant section of your website once you've constructed it.

There's a quick start guide at magpi.cc/myfaveceleb for first-time bloggers, and an introduction to creating websites at magpi.cc/html&css. For something a little different, you can create a more detailed online magazine (magpi.cc/htmlmagazine) – an ideal independent learning project for secondary school students that will encourage more detailed research, as well as web design and layout creativity.

If making websites proves rewarding, you can take things a step further with the detailed guide to hosting your own site at magpi.cc/lamp – perfect for anyone keen on a career in web development and design.



▲ Students can learn LAMP (Linux, Apache, MySQL, PHP) skills and build a fully working website

PHOTOGRAPHY



▲ Time-lapse photography captures changes throughout the day or year



Stop-motion animation and photography are two great ways to make use of Raspberry Pi. Automating the recording of a scene over a period of time is ideal for environmental projects such as monitoring a plant or insect's growth, capturing how much traffic passes your house at different times of the day, or the changing light levels.

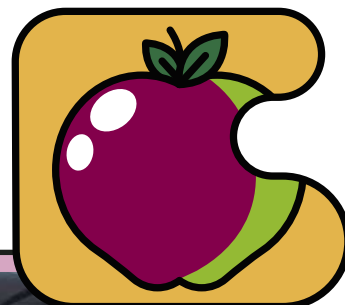
Raspberry Pi sells a High-Quality Camera board that connects directly to the CSI (camera serial interface) on Raspberry Pi (magpi.cc/hqcamera).

This opens up a range of camera-based projects (magpi.cc/camprojects). There's a particularly good introduction to how time-lapse photography works on the Raspberry Pi projects website (magpi.cc/timelapse). Should you decide to delve further into the world of Raspberry Pi photography, you might want to take your camera out and about, in which

case investing in a compact Raspberry Pi Zero might make sense. The official case even comes with a small camera hole, making for a small, self-contained camera.

Pimoroni has a detailed guide to time-lapse photography using an OctoCam. Controlled remotely, it enables you to take your project out and about and set it up almost anywhere, perhaps even recording how the landscape changes over an entire season (magpi.cc/octocamtimelapse).

“ Stop-motion animation and photography are two great ways to make use of Raspberry Pi ”



DESIGN AND ART

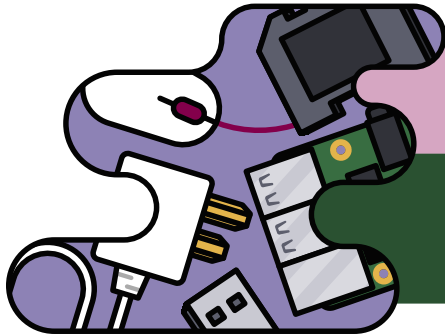
If you haven't already discovered the weekly Digital Making At Home YouTube broadcasts (rpf.io/home) put together by our Raspberry Pi US colleagues, we can certainly recommend them. Packed with inspiring ideas for things to make, the online programme aimed at young coders often features arty projects (there are several at magpi.cc/letsmakeart).

Choose Art in the drop-down menu of the Raspberry Pi Projects page to access a dozen interesting creative options. We really like the Python/CSS-based pixel art generator: magpi.cc/pixelart. This involves writing simple programs and is on the Digital Making Curriculum. In this course, students learn how to specify colours for a pixel grid, and eventually customise the display using any recognisable shades. Once you've



▲ Use code to create art in a browser or with a Sense HAT

learned how to select colours, try out the rainbow Sense HAT tutorial (magpi.cc/rainbow) and generate pretty LED patterns and scrolling messages.



MAKE WITH CODE

Physical computing is where Raspberry Pi is at. And few things help school learning as much as making something fun. Here are some great projects for students to explore

IMPACT FORCE MONITOR

Just how many Gs does that gymnastics routine involve, and how hardcore is that super-scary new roller-coaster? SparkFun's Helmet Impact Force Monitor uses Raspberry Pi Zero to investigate just how dangerous adrenaline-inducing activities can be. Once your helmet monitor is ready to use, try it out next time you're cycling or skating, or lash it to a skateboard or something else that's sturdy enough to withstand a deliberate crash. A great way to get the message about the importance of helmets across, it's also an excellent physics experiment. Note that we don't recommend this project for anyone who tends to worry.

magpi.cc/impactforce



▲ Just how much impact does crash-landing involve? The Helmet Impact project finds out

AMAZING IMAGE IDENTIFIER

It's fascinating to explore the world around us and share enthusiasm for it with others. AI (artificial intelligence) can be helpful in recognising and tagging items that crop up repeatedly, which is exactly what the Amazing Image Identifier does! Using machine learning, pupils methodically photograph and log items from nature, a museum, or other defined space, and then add interesting details, much like a curator. Executing code for an image classification program, they can then link their findings with details about it.

magpi.cc/imageidentifier



▲ Use artificial intelligence to identify, tag, and describe wildlife along with intriguing artefacts

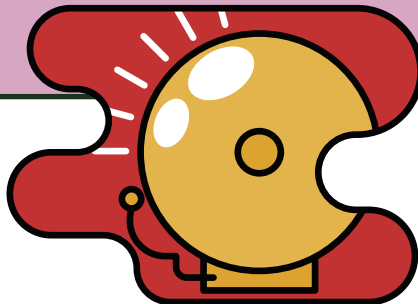
MATRIX SAND EMULATOR

Geographers and geologists can get a feel for how sand behaves and learn some physics along the way. This handheld simulator makes use of an accelerometer and a Matrix bonnet on top of Raspberry Pi. With this build, pupils get a feel for how velocity and collisions affect how the sand particles respond. They also learn how to control Raspberry Pi remotely using SSH, an invaluable coding skill they'll be able to implement in dozens of future projects of their own design. Depending on the equipment available at school, you could also 3D-print the handles, making it easier for pupils to twist and turn the Sand Emulator.

magpi.cc/matrixsandtoy



▲ Applied physics and geography in the form of a sand movement emulator



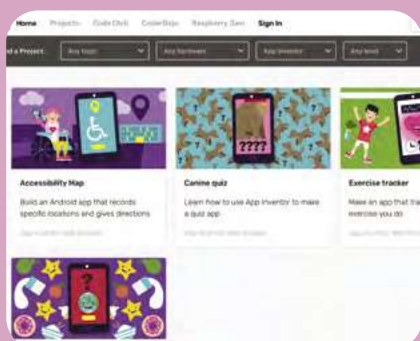


“ What do you think you could do with Raspberry Pi? ”

APP INVENTOR FOR SOCIAL ENTERPRISE

Citizenship and PSHE lessons encourage students to think about others' experiences of the world. These citizen science projects encourage them to put their ideas into action, building Android apps for phones and then logging places that are or aren't wheelchair-accessible, for example. By adapting these very same smartphone Scratch coding skills, they can also undertake a 'where does my food come from' project looking at food miles and global agriculture. The next step, of course, is to conjure up an app idea themselves and then create it.

magpi.cc/appinventor



▲ Create a citizen science smartphone app

HISTORIC VOICEBOT

Interrogating historical figures is a great way to bring them to life through role-play, and scores pretty highly as an effective learning method. This fun approach sees kids identify a worthy figure to study and gen up on them, then work through a Dialogflow chart to understand how their Historic Voicebot connects, learning some rather cool Node.js programming along the way. Having hooked up their chatbot with a touchscreen and user-friendly buttons, they can take turns interrogating their chosen subject in a way that's not at all reminiscent of Bill & Ted. At all.

magpi.cc/historicvoicebot



▲ Ask Ada Lovelace anything with the Historic Voicebot

DESIGN YOUR OWN PROJECT

This last project is down to you: what do you think you could do with Raspberry Pi? What do you think it's most worthwhile for you to focus coding skills on? We've given you loads of ideas, so now we want you to invent something of your own. More of a writer than a maker? How about writing up a detailed report on a project you've found and followed? Since you've spent time focusing on lots of different STEM and creative projects, perhaps you could write up your workings and post them online to inspire other makers. Let us know at *The MagPi* if you've made something you're really proud of – we love sharing project ideas with our readers, and they often inspire other makers. You can email us at magpi@raspberrypi.com.



Pico Wireless Pack

SPECS

PROCESSOR:
ESP32-WROOM-32E module

CONNECTIVITY:
WiFi
802.11b/g/n,
2.4GHz

FEATURES:
RGB LED,
microSD card
slot, tactile
button

► Pimoroni ► magpi.cc/picowireless ► £12 / \$14

Add wireless networking to your Pico projects. By **Phil King**

While Raspberry Pi Pico is an inexpensive yet powerful microcontroller, one drawback for IoT projects is the lack of built-in wireless LAN.

That's where the Pico Wireless Pack comes in, enabling you to connect Pico to your WiFi network over the 2.4GHz band. The Pack's wireless connectivity is provided by an on-board ESP32 MCU module via SPI.

“The obvious downside is that it blocks access to Pico's GPIO pins”

In addition, the board features a bright on-board RGB LED, tiny tactile switch, and a microSD slot for storing extra data, expanding greatly on Pico's 2MB of flash storage.

To connect your Pico, you'll need to have male headers soldered on, then you can just plug it into the female headers of the Wireless Pack, so it and Pico are back to back.

The obvious downside is that it blocks access to Pico's GPIO pins. So, to connect sensors or other

components, you'll need a stacking header on Pico or a breakout board such as Pimoroni's Pico Omnibus. Still, 14 GPIO pins are left unused; plenty for most projects.

Software libraries

Libraries for C++ and MicroPython can be downloaded from Pimoroni's GitHub repo. The latest version of Pimoroni's MicroPython UF2 firmware contains the necessary drivers, so you'll need to flash Pico with it.

To help you get to grips with the picowireless library, a couple of MicroPython examples are provided. One creates a simple web server enabling you set the red, green, and blue values of the on-board LED. The other demonstrates connecting to an API, and lets you control the LED's colour using #cheerlights tweets.

Both examples are pretty verbose, with some complex HTTP request details, so are hard to follow unless you know your networking, but you could easily adapt them for your own projects.

Alternatively, you can code with CircuitPython, using Adafruit's CircuitPython and ESP32 libraries, by adjusting a few pin numbers in their code examples. <https://magpi.cc/picowirelessjetpack>

► Adding wireless connectivity to Pico, the Wireless Pack also features an RGB LED, tactile button, and microSD slot for extra storage

Verdict

A simple way to add wireless connectivity to Pico, although you'll need to break out the GPIO pins for IoT projects.

8/10





It's official! Name change announcement!
Same team, same company - just a new name!



PiShop.ca

Legendary  **BuyaPi.ca**

is now  **PiShop.ca**

The same great products and services
From the growing PiShop family!
Now serving all of North America

PiShop.ca • PiShop.us • PiShop.mx



BuyaPi.ca

HyperPixel 2.1 Round

SPECS

DISPLAY:

2.1" IPS, 480×480 pixels, 18-bit colour, 60 fps frame rate

TOUCH

INTERFACE:

Capacitive touchscreen with multi-touch

BREAKOUT:

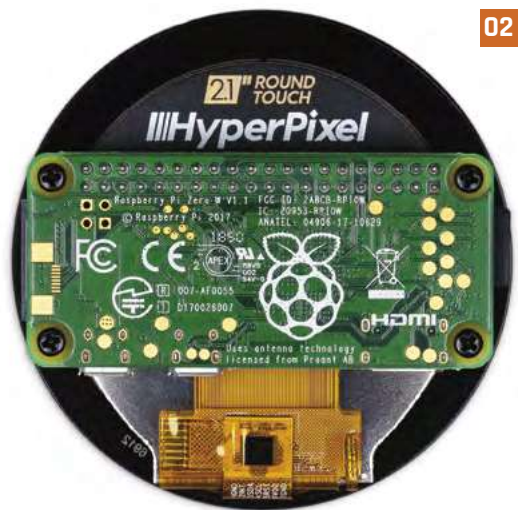
Five-pin breakout header using alternate I2C

► Pimoroni ► magpi.cc/hyperpixel2r ► £48 / \$56

This high-quality circular display doesn't cut any corners. By **Phil King**



01



02

- 01. The screen is very vivid and can be oriented how you like – a terminal command is used to rotate the display output
- 02. A Raspberry Pi Zero fits neatly on the rear, which also features a five-pin breakout header (obscured)

Circular displays are all the rage in items like smart thermostats and high-end coffee machines. The HyperPixel 2.1 Round is the first one available for Raspberry Pi.

Essentially, it's a square 480×480 pixel display minus the corners. The bezel is a little wide at around 9 mm, but the touchscreen display itself is impressive. Like the standard HyperPixel, it uses a high-speed DPI interface, allowing it to shift five times more pixel data than the SPI interface typically used for other small screens.

A Raspberry Pi Zero fits neatly onto the rear, although you'll need to be careful when mounting it not to crack the screen – it's best to put it face down on a soft surface. Short standoffs (not supplied) can be used to secure it. If using a full-size Raspberry Pi, you'll need a GPIO booster header.

While the HyperPixel 2.1 Round uses all the GPIO pins, five breakout pins on the rear provide the option of connecting sensors via an alternate I2C interface.

Software libraries

Libraries for C++ and MicroPython can be downloaded from Pimoroni's GitHub repo. The

latest version of Pimoroni's MicroPython UF2 firmware contains the necessary drivers, so you'll need to flash Pico with it.

“ The obvious downside is that it blocks access to Pico's GPIO pins ”

To help you get to grips with the picowireless library, a couple of MicroPython examples are provided. One creates a simple web server enabling you set the red, green, and blue values of the on-board LED. The other demonstrates connecting to an API and lets you control the LED's colour using #cheerlights tweets.

Both examples are pretty verbose, with some complex HTTP request details, so are hard to follow unless you know your networking, but you could easily adapt them for your own projects.

Alternatively, you can code with CircuitPython, using Adafruit's CircuitPython and ESP32 libraries, by adjusting a few pin numbers in their code examples. [\[7\]](#)

Verdict

If you really want a round display, maybe for a Halloween animated eyeball, this is an excellent option.

8/10

pi-top [4]



ROBOTICS KIT

Robotics & rapid prototyping with your Raspberry Pi

Power your projects with computer vision and applied AI

pi-top [4] Robotics Kit comes with electronic components such as a wide-angle camera, servos and motors, all of which plug and play with the pi-top [4] Complete or pi-top [4] DIY Edition*.

pi-top Robotics Kit with Expansion Plate
£187.90 / \$199.90



Gesture
Control



Obstacle
Avoidance



Autonomous
Driving



Object
Recognition



Emotion
Mapping



Line
Recognition



Interaction



Face
Tracking



pi-top.com/MagPi

Raspberry Pi is a trademark of the Raspberry Pi Foundation. *pi-top [4] and Robotics Kit with Expansion Plate sold separately.

© CEED Ltd. 2021

pi-top

Raspberry Pi made simple, robust and modular.

Air Quality Kit for Raspberry Pi

SPECS

CONTROLLER:

ATtiny1614 microcontroller

INPUT /

OUTPUT:

CCS811 digital gas sensor, 6 × LEDs, FUET-1230 piezo buzzer, 40-pin GPIO connector

COMPONENTS:

Air Quality Board, jumper leads, Raspberry Leaf GPIO template

► Monk Makes ► magpi.cc/airqualitykit ► £23 / \$32

Teach kids to get an electronic nose for pollution with this easy-to-use kit. By **Lucy Hattersley**

This Air Quality Kit board is a new addition for Raspberry Pi 400 that measures the quality of air in a room and the temperature. The kit can also be used with other Raspberry Pi computers via the GPIO pins using the supplied jumper cables and the supplied Raspberry Leaf GPIO template.


“The Air Quality Kit starts working the instant you plug it in”

It's simple to set up. Thanks to an ATTINY1614 microcontroller (not RP2040, unfortunately). Air Quality Kit starts working the instant you plug it in

and power it up. The six LEDs on the board (green, orange, and red) light up to indicate the air quality in the room.

Fresh air

Things get more interesting when you start to hack into the board. It uses a CCS811 gas sensor (magpi.cc/ccs811) to measure air quality and also packs a FUET-1230 piezo buzzer to use for audio alarms.

Moving beyond the plug-and-play aspect, you can code the Air Quality Board with Python. Code and example projects are available on GitHub (magpi.cc/aqgit). The Air Quality Board instructions (magpi.cc/aqkitpdf) walk students through setup and coding. You will need to install guizero (magpi.cc/guizero) to use some of the example code, but the results are fun and interactive. 

Verdict

A great kit for experimenting with monitoring air quality. Very easy to set up and start using, and with some good coding examples and excellent tutorials.

8/10



◀ The Air Quality Kit board connects directly to Raspberry Pi 400 and works as soon as you power it up

Wireframe

Join us as we lift the lid
on video games




Visit wfmag.cc to learn more

10 Amazing:

Game development projects

Making games on and for Raspberry Pi can be a lot of fun

Making a video game is the dream of many young (and some older) people, and it's never been easier to do so, especially when you throw a Raspberry Pi into the mix. Whether you're making something entirely digital, or games with a real-world connection, Raspberry Pi can be your gateway to game dev. 



▲ Quiz game

Recycling buzzers

These are more controllers than games, but they can be used to create a centralised buzzer system for your next pub quiz party. Everyone does those, right?

magpi.cc/quizgame



▲ Digital Making at Home: games

Foundation fun

One of the themes from the Raspberry Pi Foundation's Digital Making at Home series of livestreams over the last 18 months. In this one, you'll learn how to make games.

magpi.cc/dmahgames

► Build a retro game with PICO-8

64-pixels dev

PICO-8 is a very limited game development program that uses extremely simple graphics, but it can be used to create some incredible, complex games.

magpi.cc/pico8game



▲ Digital Making at Home: retro games

8-bit fun

Similar to when the team made games, this time they're looking at pixels, sprites, and screens with some retro game development ideas.

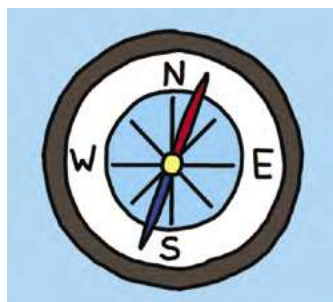
magpi.cc/dmahretro

► Compass maze

Magnetic navigation

This digital game is controlled using a Sense HAT that acts as a compass, turning in real life as you point it in different directions.

magpi.cc/compassmaze



▲ Flappy Astronaut

Avian clone

Similar to the once popular Flappy Bird, this version has you shake a Sense HAT to move your astronaut through a hazardous maze of deadly green pipes.

magpi.cc/flappyastro



▲ Lights out!

Fist of the North LED

Test your reaction time by pressing the button related to an LED that has just lit up at random. If you're too slow, you've already lost.

magpi.cc/lightstout



▲ MCPI Revival

New Minecraft Pi

Minecraft Pi on Raspberry Pi is an excellent way to learn about game programming as you hack it with simple Python code. This Revival/Reborn edition aims to update it.

magpi.cc/mcpirevival



▲ Robot Pac-Man

Real-life VS

When we say 'Pac-Man in 3D', we don't mean 3D graphics – we mean 3D-printed robots that can munch on virtual pellets and chase you down.

magpi.cc/robotpacman



▲ Pixel pet

Digital monster

Create your own virtual pet right on your Sense HAT and take good care of it with walks and words of affirmation.

magpi.cc/pixelpet

Learn Python with Raspberry Pi

Start coding in Python using a Raspberry Pi and these resources. By **Phil King**

Learn to Code by Solving Problems

AUTHOR

Daniel Zingaro

Price:
£30 / \$35

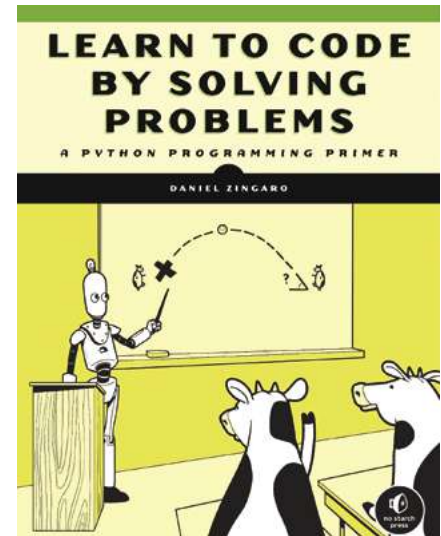
[magpi.cc/
zingaropython](http://magpi.cc/zingaropython)


Python is one of the most in-demand programming languages in the world, used to write web apps, games, machine learning software, and more. Since it's more structured and readable than some other languages, Python is also ideal for learning to code.

Suitable for beginners, Daniel Zingaro's book takes you right from the start of your Python programming journey and

focuses on core coding concepts such as loops, lists, sets, functions, file management, dictionaries, and algorithms. It finishes with a chapter on how to write more efficient code.

More than just Python, you'll learn the type of thinking required to solve problems using a computer. The key is to use top-down design to break



a larger problem down into smaller tasks that can be tackled using functions. To this end, the book is based on solving problems found on three online programming judge sites, so you can use them to check if you've got the correct solutions. 

Free online resources

Learn Python on the web for free



CODE CLUB: PYTHON MODULE 1

The first of two modules in Code Club's Python course, it introduces the basic elements via a series of fun projects including a turtle racing game and secret messages program.
► magpi.cc/python1

PROGRAMMING FOR EVERYONE

Created by the University of Michigan, this highly rated Python course breaks

everything down into easy-to-digest chunks and will give you a thorough grounding in the language.

► coursera.org/learn/python

LEARNPYTHON.ORG

Featuring a built-in code editor and interactive exercises to complete, this site covers all the Python basics along with more advanced topics, such as regular expressions and decorators.

► learnpython.org

Programming the Raspberry Pi, Third Edition

AUTHOR

Simon Monk

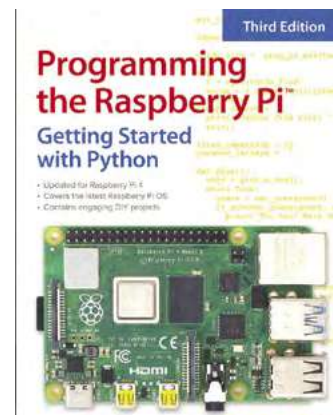
Price:
£13 / \$15

magpi.cc/monkpython

If you're after a Raspberry Pi-focused guide to programming in Python, this newly released and fully updated third edition of Simon Monk's tome is just the ticket. After an introduction to setting up your Raspberry Pi and exploring the desktop (skip this if you're already familiar), it moves on to Python programming. Interestingly, it uses the Mu code editor instead of the default Thonny IDE.

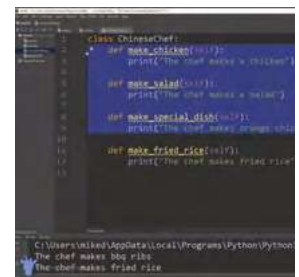
After exploring Python core concepts such as conditionals, strings, functions, modules, and classes, it moves on to cover graphical user interfaces (with the guizero module), games programming (with Pygame),

and interfacing with hardware using the GPIO pins. The last topic is expanded upon with three projects to build, including an autonomous robot. [M](#)



Video tutorials

Learn to code in Python by watching YouTube videos



FULL COURSE FOR BEGINNERS

One of the many free programming resources by freeCodeCamp.org, this four-hour video takes you through a whole bunch of key concepts in Python.

► magpi.cc/fccpythonyt

LEARN PYTHON IN AN HOUR

Can you really learn Python in an hour? Programming guru Mosh gives it a good go, taking you through all the basics, from variables and operators to loops and tuples.

► magpi.cc/mosh1hrpython

LEARN PYTHON IN FIVE HOURS

Presented by Nana Janashia, this marathon video (split into time-stamped sections) explains all the core concepts in detail before moving on to advanced elements such as object-oriented programming and using APIs.

► magpi.cc/nanapython

Learn Python 3

AUTHOR

Codecademy

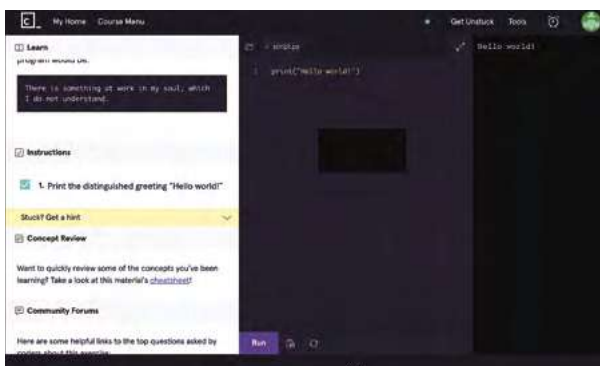
Price:
£16 / \$20 per month

magpi.cc/codecademypython

While many of Codecademy's learning resources (including Python 2) are free, this new introductory Python 3 course requires a Pro membership to access. Along with written

instructions, the browser-based interactive environment features a code editor so you can type in and run your Python listings on the fly, with the output shown on the right. You also have access to Codecademy forums to ask questions and discuss topics – useful if you get stuck.

Taking an estimated 25 hours to complete, this comprehensive course takes you from printing 'Hello World' to using Python classes, via conditionals, lists, loops, strings, functions, modules, dictionaries, and files. There are also optional code challenges to test your newfound knowledge. You'll get a certificate at the end, too. [M](#)





Geeky Faye

An artist and maker, Geeky Faye describes herself as a one-man band, tackling whole areas of creation

- Name **Allie aka Geeky Faye** | ➤ Occupation **YouTuber**
- Community role **Creator** | ➤ Website **magpi.cc/geekyfaye / @GeekyFayeArt**

Having multiple hobbies and interests can be fun, but they can sometimes get on top of you. Allie, also known online as Geeky Faye, seems to have thrived with so many. “As it currently stands, I will happily refer to myself as a maker, artist, designer, and filmmaker because all of those are quite accurate to describe the stuff I do!” Allie tells us. “I’ve been making almost my

whole life. I dove headlong into art as a young teen, to be quickly followed by cosplay and building things that I needed for myself. I would go on to get a degree in fine arts and pursue a professional career as an artist, but that actually ended out resulting in me being on a computer all day more than anything! I’ve always needed to use my hands to create, which is why I’ve always been drawn to

picking up as many making skills as possible... These days my making is all very ‘multimedia’ so to speak, involving 3D printing, textiles, electronics, wood working, digital design, and lots of paint!”

When did you learn about Raspberry Pi?

I’d heard about Raspberry Pi years ago, but I didn’t *really* learn about it until a few years back when I started getting into 3D printing and discovered that you could use one to act as a remote controller for the printer. That felt like an amazing use for a tool I had previously never gotten involved with, but once I started to use them for that, I became more curious and started learning a bit more about them. I’m still quite a Raspberry Pi novice and I am continually blown away by what they are capable of.

What have you made with Raspberry Pi?

I am actually working on my first ever proper Raspberry Pi project as we speak! Previously I have only set them up for use with OctoPrint, 3D-printed them

▼ The PCB for this project was custom-made by Geeky Faye



“I am actually working on my first ever proper Raspberry Pi project as we speak!”

Showing the process

“One of the things I’m actually trying to do in sharing all my experiences so publicly is to inspire others to explore and try new things as well. The internet does a *really* good job of convincing us that everyone is an expert the first time they do something, which can be so demoralising to someone wanting to try something for the first time. That’s why I’m making such an effort to show myself learning new things alongside doing the stuff I have more experience at.”

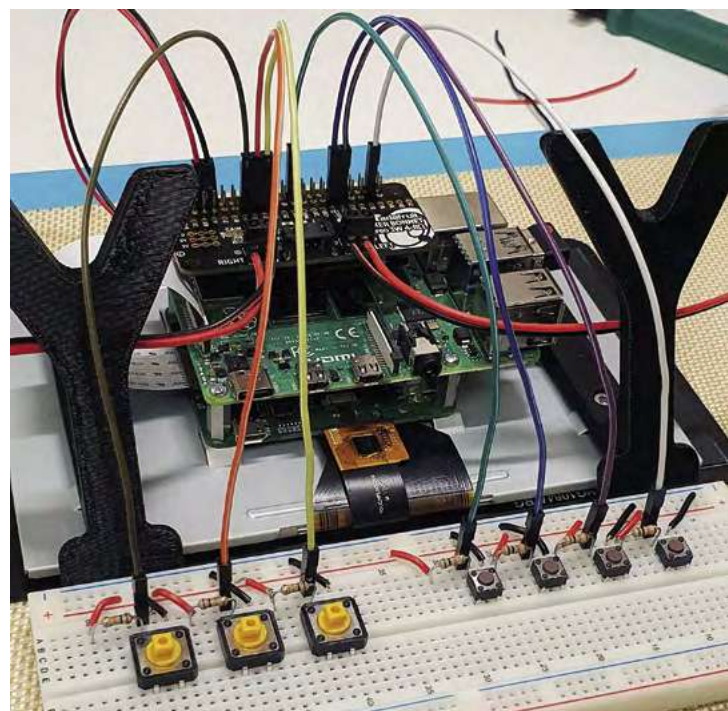
▼ Geeky Faye’s amazing Hange Zoë cosplay from the *Attack on Titan* series



a case, and then let them do their thing. Starting from that base need, I decided to take an OctoPrint server [Raspberry Pi to the next level and started creating BMOctoPrint; an OctoPrint server in the body of a BMO (from *Adventure Time*). Of course, it would be boring to just slap a Raspberry Pi inside a BMO-shaped case and call it a day. So, in spite of zero prior experience (I’m even new to electronics in general), I decided to add in functionality like physical buttons that correspond to printer commands, a touchscreen to control OctoPrint (or anything on Raspberry Pi) directly, speakers for sound, and of course user-triggered animations to bring BMO to life... I even ended out designing a custom PCB for the project, which makes the whole thing so clean and straightforward.

What’s your favourite project that you’ve done?

Most recently I redesigned my teleprompter for the third time and I’m finally *really* happy with it. It is 3D-printable, prints in



just two pieces that assemble with a bit of glue, and is usable with most kinds of lens adapters that you can buy off the internet along with a bit of cheap plastic for the ‘glass’. It is small, easy to use, and will work with any of my six camera lenses; a problem that the previous teleprompter struggled with! That said, I still think my modular picture frame is one of the coolest, hackiest things that I’ve made. I highly recommend anyone who frames more than a single thing over the course of their lives to pick up the files, as you will basically never need to buy a picture frame again, and that’s pretty awesome, I think. *M*

▲ Testing out various functions of BMOctoPrint



◀ Photography, 3D-printing, and making come together in another project

This Month in Raspberry Pi

Astro Pi winners

More award-winning projects from space

Last issue we showed you some highlights from the recent Astro Pi Mission Space Lab. Since then, the winners for that competition were announced, with some incredible experiments to show for their efforts. [M](#)



Zeus

Tudor Vianu National College of Computer Science

📍 Romania

► magpi.cc/astrozeus

With the theme of Life on Earth, the team used previous Astro Pi data along with IR photos of clouds to try and predict weather patterns.



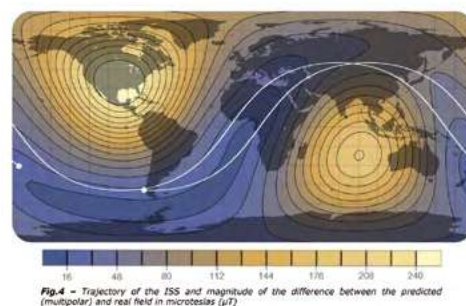
Atlantes

Niubit Coding Club

📍 Spain

► magpi.cc/astroatlantes

A fun project that uses weather data from the location of the ISS and sensor data from Astro Pi to create music using sonification.



Mag-AZ

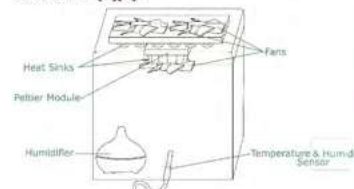
Escola Secundária Domingos Rebelo

📍 Portugal

► magpi.cc/astromagaz

The Portuguese team studied the Earth's magnetic field to create a model of it, including the poles. Their results were a little off, but it was an interesting experiment nonetheless.

We built a temperature and humidity control room, by combining two projects found online [3][4].



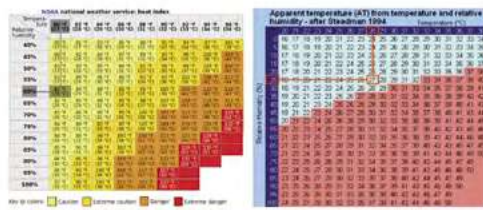
Mateii

Saint Sava National College

📍 Romania

► magpi.cc/astromateii

A Life in Space project that created a predictive model for mould growth in space, specifically ones used in food and medicine production.



Juno

Institut d'Altafulla

📍 Spain

► magpi.cc/astrojuno

How hot is it really on the ISS? Using temperature and humidity data, this team was able to determine how hot it feels, and that's apparently 27°C.

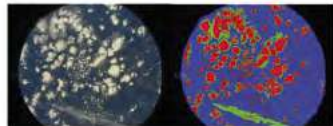


Illustration 1: on the left the original picture, on the right the treated picture with sea in blue (45%), clouds in red (12%) and land in green (39%).

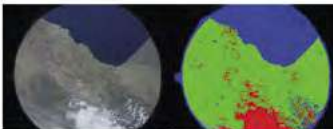


Illustration 2: on the left the original picture, on the right the treated picture with sea in blue (22%), clouds in red (8%) and land in green (43%).

Albedo

Lycée Albert Camus

📍 France

► magpi.cc/astroalbedo

Albedo refers to the amount of solar radiation a body, like the Earth, reflects/absorbs. This project measured the value of Earth's albedo in relation to global warming.



SpaceRad

Centrum Nauki Kopernika – Planetarium Venus

📍 Poland

► magpi.cc/astrospacerad

Another project measuring albedo, although this version was in search of locations to place solar panels for the best energy generation.



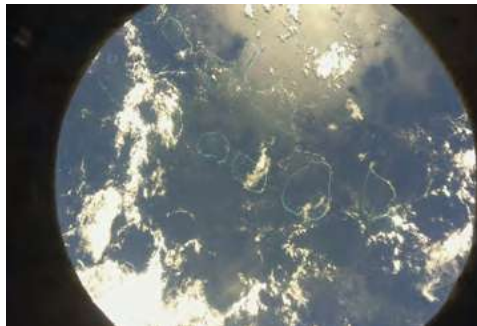
Magtrix

The Leys School

📍 UK

► magpi.cc/astromagtrix

This magnetic field test was investigating how the Earth's magnetic field changes throughout the day, and how it may be shaped by solar winds.



Mechabot

Robone Robotics Club

📍 Germany

► magpi.cc/astromechabot

A Life on Earth project that tested how magnetic fields may change/have changed depending on climate, and also if they affect objects in Low Earth Orbit.



SPACEi2

Zanneio Model High School

📍 Greece

► magpi.cc/spacepi2

Using the IR camera on Izzy the Astro Pi, vegetation health was checked against older data. In addition, the magnetic field readings were recorded.

Check out the full blog about the winners, and some runners-up, at: magpi.cc/astro21blog

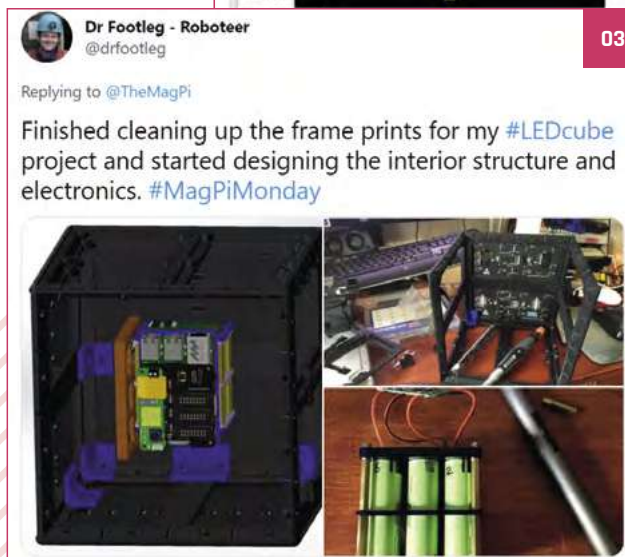
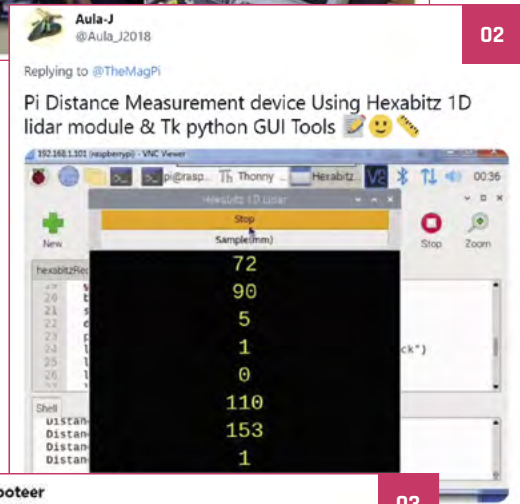
MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!!

01. Doofus is taking shape – maybe they should take part in the next Pi Wars?
02. This is a cool bit of code using lidar to measure distance!
03. We look forward to seeing the final version of this great-looking LED cube
04. A very neat and accessible project that can easily be modified to Braille for blind users
05. Robot experiments are just what we like to see on a Monday morning
06. A way to safely peruse museums if you're still not keen on going out
07. We're not entirely sure what all these words mean, but it looks and sounds neat
08. Pico stream decks are going to definitely become a thing – we're already using one ourselves
09. Play a physical pinball machine over the internet? Not as strange as you think
10. An incredibly useful project, we'll take ten
11. A classic, and quite practical, cluster





CycOb
@cyc_ob

05

Replying to @TheMagPi

#ROS2 on #GoPiGo3 #RaspberryPi 3B+ #Robot Dave (and on #Ubuntu 20.04 on #VMwareFusion on #MacMini)



Modular Robotics and Robot Operating System (ROS)

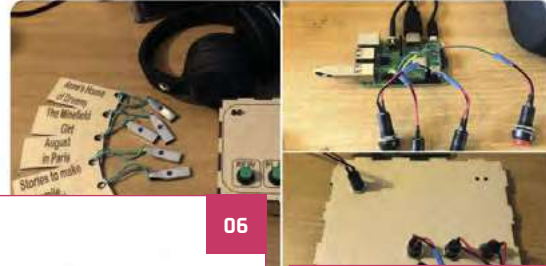


David Pride
@davejavupride

04

Replying to @TheMagPi

Final version of an accessible audiobook player (for my mum, whose eyesight is now failing) Using @Raspberry_Pi and @OpenAudible (fully copyright compliant!) Each USB stick is a single book, player can bookmark and restart from same point. #MagPiMonday

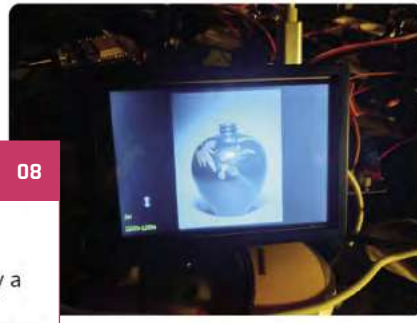


Razvan T. Coloja
@cypresswist

06

Replying to @TheMagPi

Using a RPi 3B+ and a TFT HAT to display 35,000 random images from the Cleveland Museum of Art.



Rob
@d4kineator

07

Replying to @TheMagPi

#MagPiMonday Coding more features for the Pitfalls microtonal tuning scale explorer music script - running on @monomies norms shield / raspberry pi 3b+

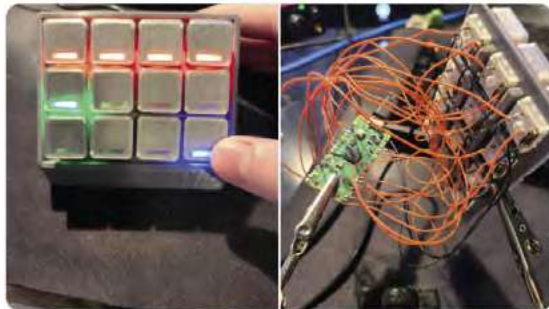


Ben Corbitt
@btcor

08

Replying to @TheMagPi

Working on a couple DIY Stream Decks as a gift to some friends who are Twitch streamers. Powered by a Raspberry Pi Pico with 3D printed enclosures and keycaps. I also used LED Sequins from @adafruit, which fit perfectly behind the key switches.



Surrogate.tv
@SurrogateTV

09

Replying to @TheMagPi

Rise & Grind! ☀️ We have been working on this internet-controlled @Raspberry_Pi powered mini-pinball game! @American_Girl & raspi working together, what a combo! 🥰



Daniel Beardsall
@Beardy

10

Replying to @TheMagPi

I prototyped a single-button keyboard to make up for a manufacturing oversight...

Daniel Beardsall @Beardy · Jul 24
Mildly bothered by how @FrizzyTom's @ASUS laptop has a Home key but no corresponding (dedicated) End key... may I introduce the world's most over-engineered End button.
#RaspberryPiPico
Show this thread



Rogelio Rodriguez
@rogelman

11

Replying to @TheMagPi

I built a raspberry pi kubernetes cluster :D



Build a raspberry pi kubernetes cluster
1. Get the hardware To build this cluster you will need the following hardware parts: 1xPoE switch (link) cost EUR 1195x 15cm ethernet cables (link) cost EUR ...
@rogelicrodriguez.net

Pico Display Reaction Game

A simple and cool game project from a young reader

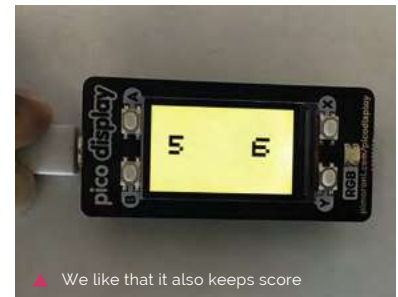
Nirvaan emailed us about this a little while ago, so we apologise for publishing it late! Here's what they had to say. "I'm Nirvaan, and I am ten years old. I live in California and I love building stuff with Raspberry Pi and Raspberry Pi Pico. I have built many projects with them so far, such as a smart doorbell, a Lego sorter, and I'm working on a robot pet named Momo.

"I have come up with an idea for a simple Pico project to be [published] in *The MagPi* magazine. My project is a Pico Display Reaction Game! I used the Raspberry Pi Pico with my Pimoroni Pico Display Pack.

"When I first got my Pico Display, I had no idea what to do with it. A month later, I came up with an idea for a reaction game using it. At first, it



▲ Who was faster?



▲ We like that it also keeps score

could only play one round. The next day, I worked on it some more and made it into a tournament. In the game, you play eleven rounds using the buttons on the display. Best of eleven wins! I have been playing with my parents and it's really fun!"

S-Pi-Dometer

A DIY speedometer for your bicycle

Apparently a lot of people got into cycling over the last couple of years, including Camilo Sanchez who emailed us about a little DIY speedometer he created called the S-Pi-Dometer.

"Keen to keep track of my rides, I thought of developing my own speedometer," Camilo says. "Raspberry Pi Zero W is the perfect tool for this project: it's fast, easy to program, and with an incorporated Bluetooth module! The principle of the speedometer is to count the number of wheel rotations and the time between them. To acquire this data, a digital Hall effect sensor module is used, then all this information is passed to Raspberry Pi and calculations are made for speed and distance, which are then handed to an Android app via serial data communication using Bluetooth RFCOMM, SPP protocol."

You can check out the source code and instructions for this project online here: magpi.cc/spidometer.



Best of the rest!

A bumper-sized best of the rest section to fit in everything from this month

LED MATRIX ALBUM ART DISPLAY



We're always amazed how good a 64 × 64 display can be from a bit of a distance. This cool project cycles through album covers broken down into 4096 coloured dots.

► magpi.cc/xZfxuq

CARDBOARD IS AWESOME



For prototyping, there's not much better than cardboard, as it's inexpensive and versatile. It's apparently also good if you need to create a Raspberry Pi robot arm.

► magpi.cc/9XNRZI

DESKTOP SOLAR SYSTEM

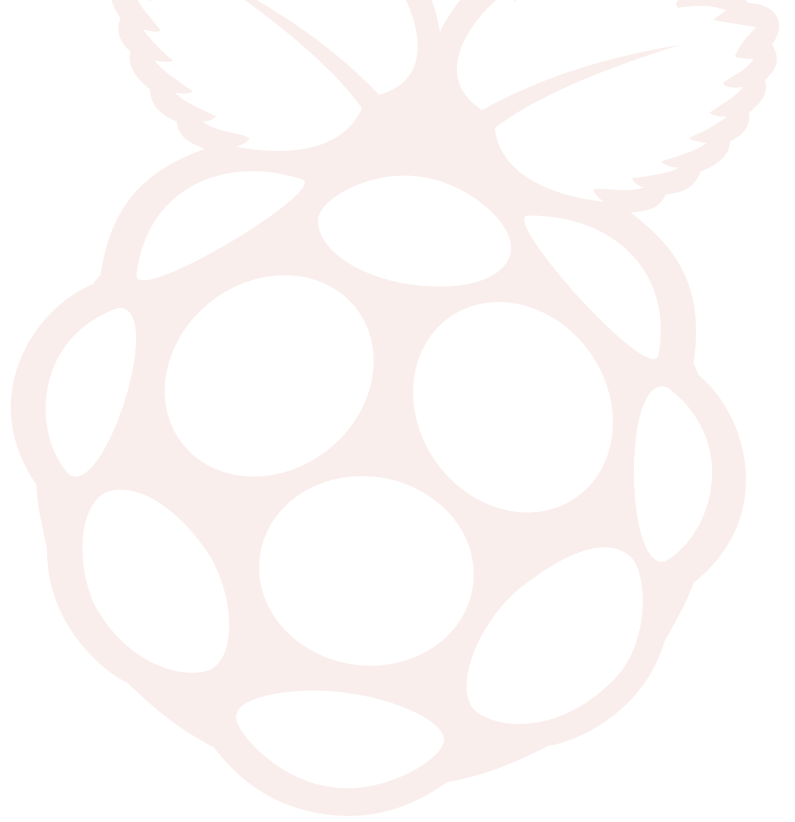
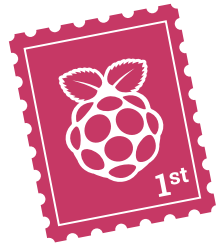


This Pico project shows you the exact placement of planets in the Solar System right now – or whenever you set the date to. It's a very smart use of a small space. [M](https://magpi.cc/fk8uwM)

► magpi.cc/fk8uwM

“ For prototyping, there's not much better than cardboard, as it's inexpensive and versatile ”

Your Letters



Android optimisations

Since Raspberry Pi 4 has an ARM processor, would it be able to run Android APKs better? Obviously won't be as fast as some of the Android phone chips, but probably better on the RAM I was thinking?

Ky via Twitter

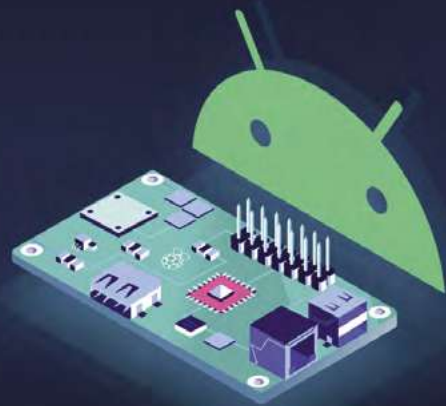
We've not seen anyone trying to run APKs natively on Raspberry Pi OS (which is Debian-based). However, there is emteria.OS, which is a build of Android for Raspberry Pi. We're currently working on an article using it, but there are a lot of optimisations that Android has for its devices, so we don't expect any Android APKs/apps to work noticeably better.

Also, Android phones are now getting to 8GB of RAM, so even in that regard, Raspberry Pi won't have more RAM for long.



Live Release of emteria.OS for Raspberry Pi 4B

The Android 11-based operating system for industrial and professional devices with a user interface is now available.



▲ If you want to try Android on Raspberry Pi, emteria.OS is the place to start



▲ The Museum of Global Communications is located near the tunnels where the first international telegraph cable landed
Image: CC BY-SA 4.0 Nilfanion

Great minds

I just read the article on the Ultimate Raspberry Pi 4 Server and glanced over at mine, built a few months ago. Same Argon40 case, same 8GB Raspberry Pi 4, smaller WD SSD (250GB). Therefore, I commend the great design chosen for this article. I'd send photos, but, except for the label on the SSD, yours looks exactly like mine.

Nate via Twitter

For our money, it's the best way to create such a server, so we're happy to see that other people agree! You can find part two of that build on page 39 – hopefully you will all approve of the extra stuff PJ has added in this part.

▼ The MagPi regular PJ Evans's new series should help you make a great server



Seeing our place in history

In her last word piece, Our Place in History, Lucy says that the majority of international internet traffic travels on submarine cables.

If you're in Cornwall, you can visit The Museum of Global Communications at Porthcurno (pkporthcurno.com).

The first international telegraph cable linking the UK and India landed in Porthcurno in 1870. Just as Lucy says, computers are a lot older than people think; so is digital communication. The 1870 telegraph cable was digital, with the signal being switched on and off, or 1 and 0 in today's language, sending messages in Morse code. Today there are fibre optic cables landing at Porthcurno carrying vast amounts of data as pulses of light. Same basic principles but billions of times faster.

And as well as an interesting museum, Porthcurno has a great beach and the Minack Theatre.

Ian via email

This looks like a very interesting museum, especially if you're into early communication! Hopefully Raspberry Pi will one day be deemed worthy of being added to the museum.

Contact us!

- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.com**
- Online **raspberrypi.org/forums**

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE
FOR THE MODERN MAKER



SUBSCRIBE AND
SAVE UP TO
35%
on the cover price



ISSUE #46
OUT NOW

hsmag.cc



Available on the
App Store



GET IT ON
Google Play

WIN ONE OF TEN

RASPBERRY PI 4 CASE FANS

The official case fan fits neatly inside an official Raspberry Pi 4 case to help improve air circulation over the main SoC. We have ten this month to give away.



Head here to enter: magpi.cc/win | Learn more: magpi.cc/casefan

Terms & Conditions

Competition opens on **25 August 2021** and closes on **30 September 2021**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

CUSTOMPC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING

THE MAGAZINE FOR

PC HARDWARE ENTHUSIASTS

DEEP DIVE **HOW NVIDIA DLSS AND AMD FSR WORK**

CUSTOMPC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING / ISSUE 217

EXCLUSIVE

BEAT THE SCALPERS

BUILD A **£1,099** PC WITH PARTS YOU CAN ACTUALLY BUY

- EVGA GEFORCE RTX 3060
- AMD RYZEN 5 5600X
- 16GB OF RAM
- 1TB PCI-E SSD
- LIQUID COOLING
- AND MORE

GROUP TESTS
CPU AIR COOLERS
WIRELESS GAMING MICE

HOW TO CLEAN YOUR WATER-COOLING LOOP

GUARANTEED FOR FIRST 100 READERS

OCTOBER 2021 / £5.99

40 YEARS OF THE PC WE TALK TO IBM

REVIEWED **NVIDIA GEFORCE RTX 3070 Ti AND 3080 Ti**

CUSTOMPC

BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING / ISSUE 216

BUILD YOUR DREAM PC

CREATE A DAZZLING SYSTEM WITH YOUR OWN PERSONAL STAMP

- WATER COOLING
- CASE CUTTING
- GLASS ETCHING
- VINYL WRAPPING
- GO SATISFYING

SPRAY-PAINT YOUR CASE

ALL HOME NETWORKING GUIDE
DOSE THE RIGHT SETUP
WIFI OR POWERLINE?
REVIEWED THE LATEST ROUTERS

SEPTEMBER 2021 / £5.99

LIQUID COOLING 240MM CPU COOLER GROUP TEST

ISSUE 217 OUT NOW

VISIT **CUSTOMPC.CO.UK** TO LEARN MORE

NEXT MONTH | *MagPi*

HALLOWEEN

FEARFULLY
GOOD
PROJECTS

THE MAGPI #110
ON SALE 30 SEPTEMBER

Plus!

Build a Pico solar system

Explore retro audio with Raspberry Pi

Learn electronics with GPIO

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER @TheMagPi

FACEBOOK fb.com/MagPiMagazine

EMAIL magpi@raspberrypi.com

EDITORIAL

Editor

Lucy Hattersley
lucy@raspberrypi.com

Features Editor

Rob Zwetsloot
rob@raspberrypi.com

Sub Editor

Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Lucy Cowan, Sam Ribbits

Illustrator

Sam Alder

CONTRIBUTORS

Mike Cook, David Crookes,
PJ Evans, Ben Everard, Martin
O'Hanlon, Rosemary Hattersley,
Nicola King, Phil King, Laura
Sach, Mark Vanstone

PUBLISHING

Publishing Director

Russell Barnes
russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd, Maurice Wilkes Building, St John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under

a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.





Making it together

Magazines are about meeting up once a month says **Lucy Hattersley**

Here at *The MagPi* magazine, we spend a fair bit of time trying to make the very best magazine we can. Every issue is packed with the best projects we can find from the amazing Raspberry Pi community, and we find amazing tutorial writers to help us get the most from Raspberry Pi.

Reading a magazine is a great way to spend your time and learn more about a hobby with other people.

I have always supported those magazines I love. I currently subscribe to Simon Brew's excellent *Film Stories* magazine, BBC Good Food, National Geographic, and I get RSPB Nature's Home and Tate Etc. delivered as a member of both fine organisations. I also have a Readly account for picking up digital magazines that I wouldn't admit to reading in public, and PressReader for picking up digital newspapers (incidentally, did you know you can read these online for free with a library card, at pressreader.com?)

You can get *The MagPi* magazine, and all of our stable of tech titles for free: HackSpace magazine, Wireframe, and Custom PC are all available as PDF downloads. This is because we want to reach as many people as possible, and make

computing as accessible as it can be to a wide range of folk. Raspberry Pi is all about lowering the barrier to entry for computing. And our magazines are a proud part of that.

Each magazine we make offers a different window into the tech and maker scene, whether it's Wireframe's developer interviews, or HackSpace's practical tutorials on wood and metal-working. I even enjoy Custom PC's deep dive into obscure PC components like cooling widgets. It keeps people off the streets, probably literally as they can't afford shoes, having spent all their money on motherboards.

Keeping it real

One thing I'm especially looking forward to is meeting *The MagPi* magazine readers again. Whether it's at Pi Wars, the Scratch party, or Raspberry Fields. It's at these kinds of events that the community comes together. This is where we really find out if it's all working. Fingers crossed we will be back attending these events soon.

In the meantime, you can really help us out by filling out our reader survey (magpi.cc/survey). We're not sure how it works at other magazines, but we set up these surveys ourselves,

" I hope you enjoy the time you spend inside these pages **"**

and the editorial team sends them to our email list.

It's not just some marketing gumpf. We read every response and take all the thoughts on board. And of course, you can always email the team directly (magpi@raspberrypi.com) or get in touch on Twitter or Facebook.

Magazines are all about people. The people featured, making the projects, and those behind the scenes beavering away to put the images and words together. Ultimately, it's all about coming together once a month to hang out and chat about our hobby, making and building with Raspberry Pi.

I hope you enjoy the time you spend inside these pages. I hope *The MagPi*, in its own small way, makes the world a better place. Even if it's just for a few people for a couple of hours every month. **"**

AUTHOR Lucy Hattersley

Lucy Hattersley is editor of *The MagPi* and is off to the pub with work colleagues for the first time in a year.

magpi.cc

CREATE FANTASTIC ELECTRONIC PROJECTS WITH FLOWCODE

There are two parts to Flowcode:

Flowcode Embedded - Allows you to create advanced microcontroller projects using graphical programming.

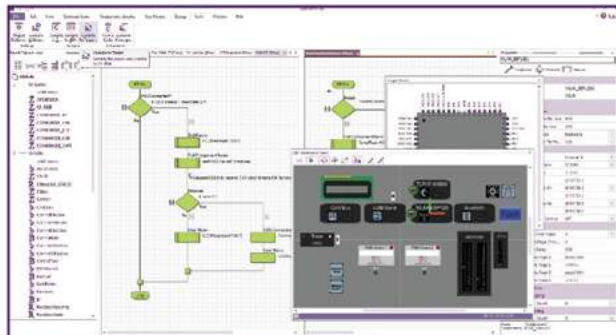
Flowcode App Developer - Allows you to create fantastic looking HMI's with low cost hardware for your Windows PC or tablet.



20% off your Flowcode purchase using code **MAGPI20**

Use Discount Code at Checkout @ www.flowcode.co.uk/buy

EMBEDDED



- Create highly functional microcontroller projects using RPi, Arduino, PIC, ESP, ARM and AVR processors
- Graphical programming: use Flowcharts, state diagrams and data flow techniques
- Full simulation - electrical and mechanical
- Huge library of parts

APP DEVELOPER



- Create great Windows Human Machine interfaces for control and data gathering using RPi, Arduino, PIC, ESP
- Graphical programming: use Flowcharts, state diagrams and data flow techniques
- Comprehensive library of dials, switches, indicators, graphs and other components

RP2040 Support

Compatible **sparkfun**
ELECTRONICS

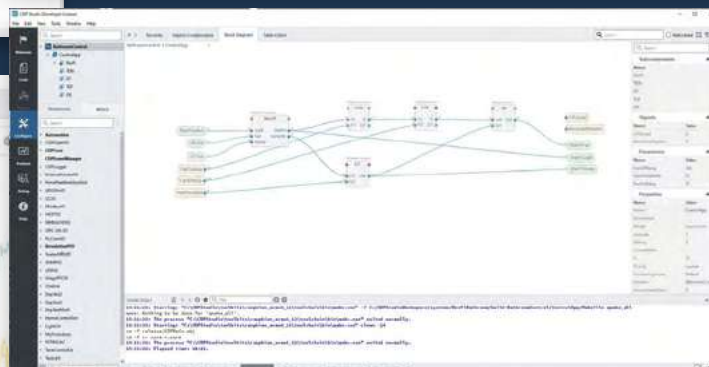
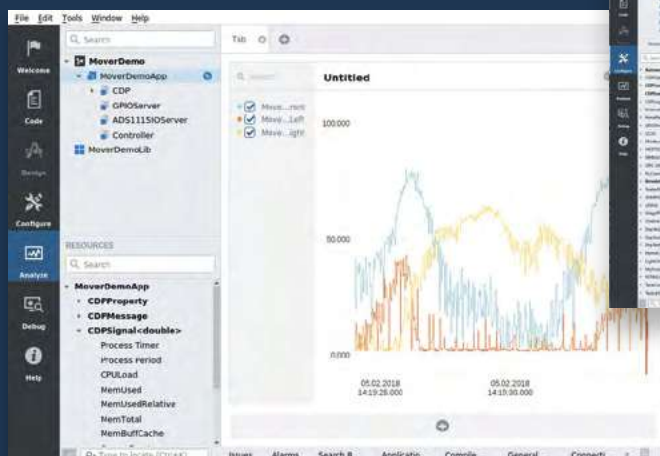
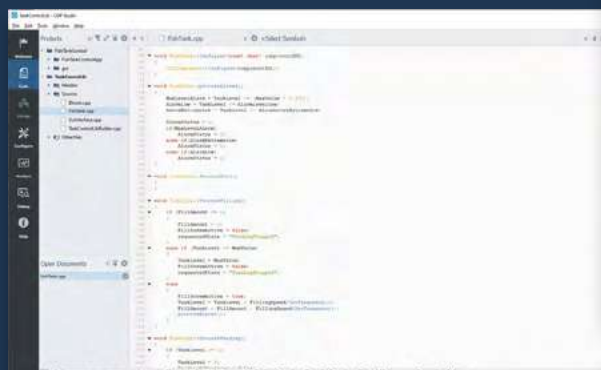


Raspberry-PI Pico



Available for **FREE** download now @ Flowcode.co.uk/download

Code



Configure

Analyze

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

CDP Studio, a great software development tool for your home projects. Build systems for Raspberry Pi, use C++ or NoCode programming, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create beautiful user interfaces. Built for industrial control system development, **FREE for home projects.**

cdpstudio.com

Tel: +47 990 80 900 • info@cdptech.com

CDP Technologies AS // Hundsværgata 8, 6008 Ålesund, Norway

